

# เอกสารประกอบการเรียนปฏิบัติการครั้งที่ 8

เรื่อง การพัฒนาฐานข้อมูล โดยการสร้าง และ อ่าน ข้อมูล  
ผ่านโปรแกรมประยุกต์ Laravel Framework



วัตถุประสงค์ : เมื่อจบบทปฏิบัติการนี้แล้วนักศึกษาสามารถ

1. เข้าใจการทำงานการสร้างโปรเจกต์โดยใช้ Laravel Framework
2. เข้าใจและสามารถใช้คำสั่งในการสร้างฐานข้อมูล
3. เข้าใจและสามารถใช้คำสั่งในการสร้างตารางและวิว
4. เข้าใจและสามารถใช้คำสั่งในการสร้างฟังก์ชันการเพิ่มข้อมูล
5. เข้าใจและสามารถใช้คำสั่งในการสร้างฟังก์ชันการแสดงผลข้อมูล

ขั้นตอนการปฏิบัติ (120 นาที)

ส่วนที่ 1

การสร้างโปรเจกต์และการตั้งค่าโปรเจกต์ Laravel

ส่วนที่ 2

การสร้างฐานข้อมูล ตาราง วิว และการเพิ่มข้อมูล

ส่วนที่ 3

การสร้างหน้าแสดงผลข้อมูล และการตกแต่งหน้าต่าง

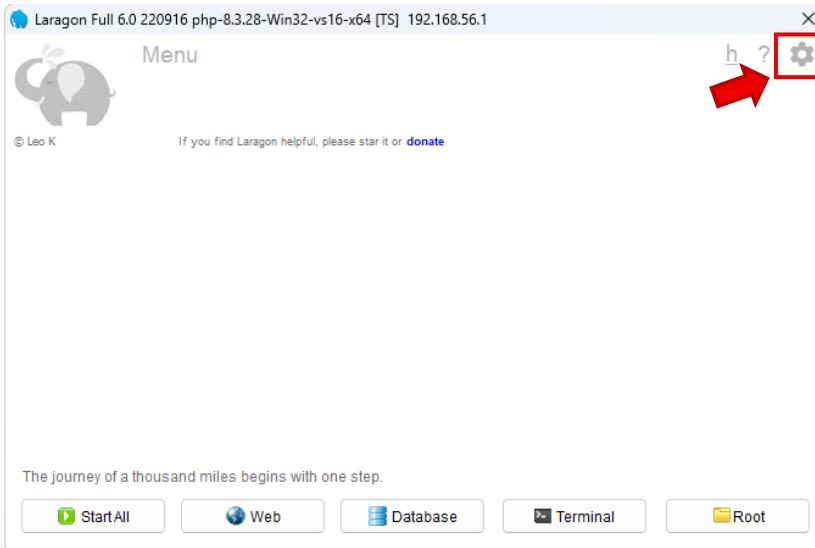
ส่วนที่ 4

การสร้างฟังก์ชันการเพิ่มข้อมูล และการตกแต่งหน้าต่าง

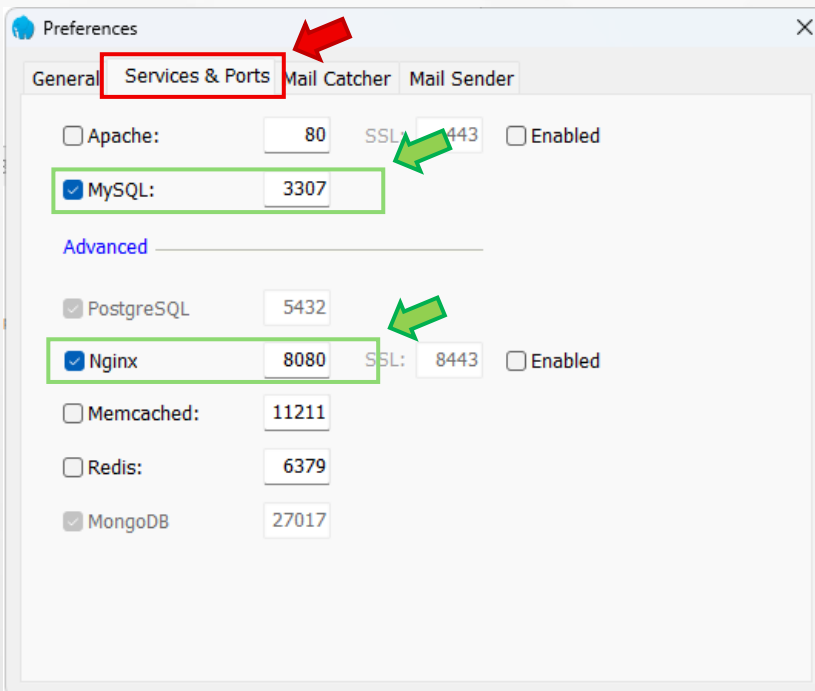
DGT01 0310 พื้นฐานการออกแบบและพัฒนาฐานข้อมูล  
สำนักวิทยาศาสตร์และเทคโนโลยี มหวิทยาลัยเทคโนโลยีสุรนารี  
ผู้ช่วยศาสตราจารย์ ดร.สฤติย์โชค โพธิ์สะอาด  
นายปิยะ รองในเมือง

## ส่วนที่ 1 การสร้างโปรเจ็คและการตั้งค่าโปรเจ็ค Laravel

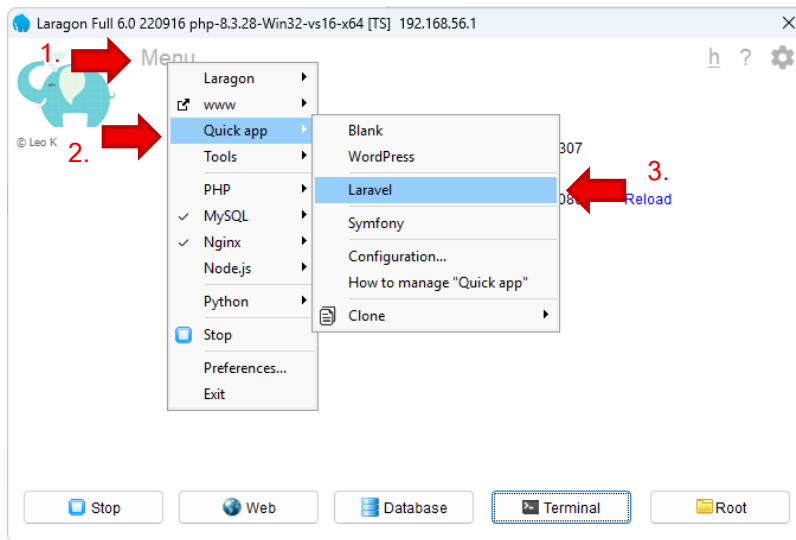
1. เปิดโปรแกรม Laragon แล้วให้ตั้งค่าตามหน้าจต่อไปนี้ เพื่อให้โปรเจ็คสามารถทำไปทิศทางเดียวกัน กดที่รูปเฟือง



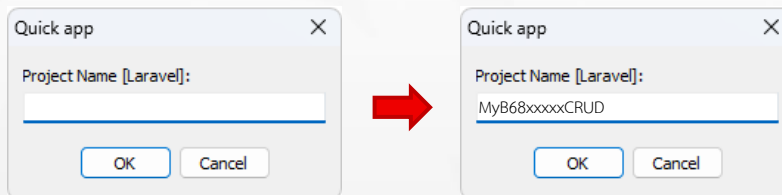
2. เลือกแท็บ Service & Ports ตามกรอบสีแดง แล้วให้กำหนด Service แล้วกำหนดเลขพอร์ตให้ตรงกับที่ล้อมกรอบสีเขียว เมื่อตั้งค่าเสร็จแล้วให้กดหน้าต่างต่างปิด จากนั้นกด Start All



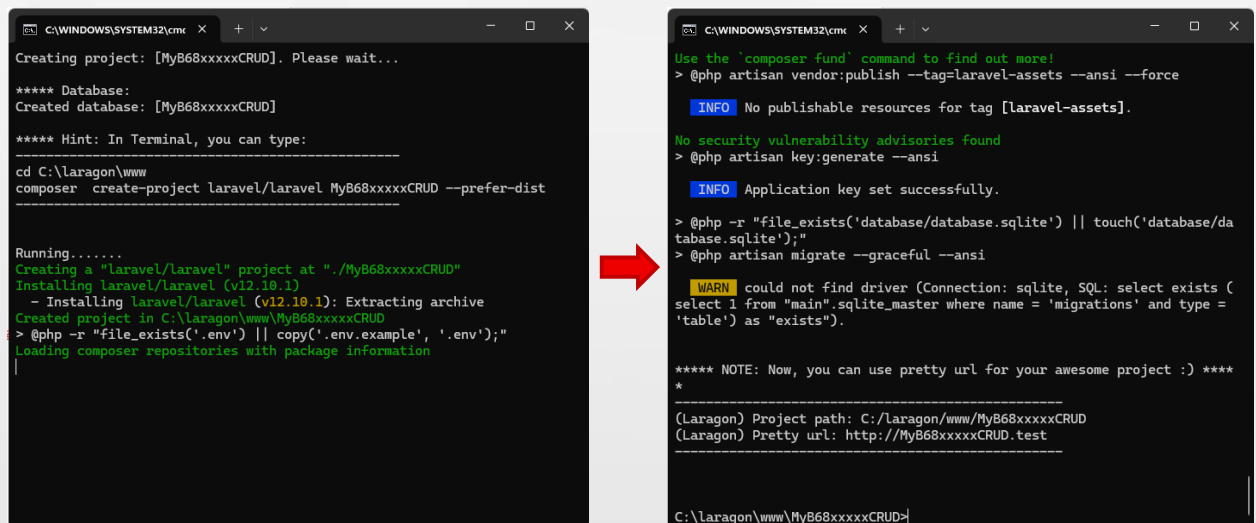
3. ให้เลือกเมนูตามขั้นตอนที่ปรากฏในรูปต่อไปนี้



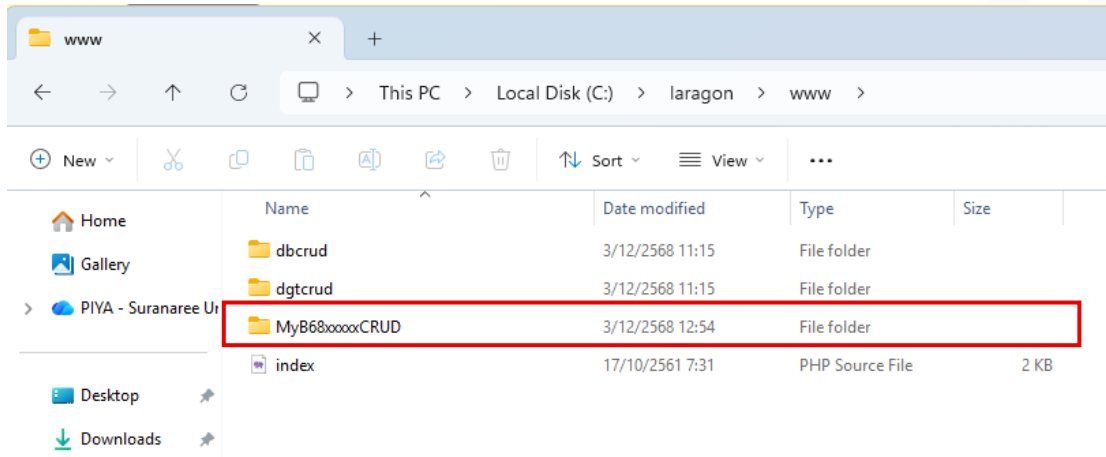
4. จะได้หน้าต่างนี้ขึ้นมา ให้ตั้งชื่อโปรเจกต์ว่า “MyB68xxxxCRUD” \*\*\*B68xxxx ให้ใส่เป็นรหัสนักศึกษาของตัวเอง แล้วกด OK เพื่อสร้างโปรเจกต์ Laravel



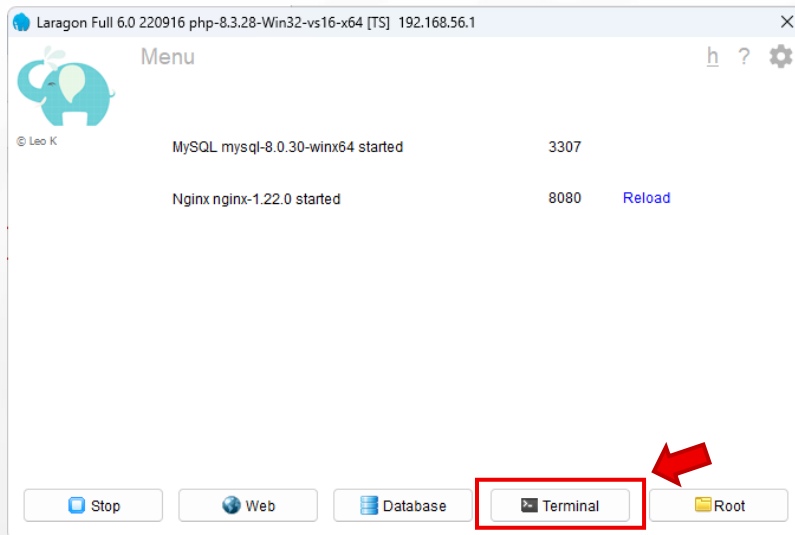
5. เมื่อกด OK แล้วจะมีหน้าต่าง Terminal ขึ้นมา ให้รอจนจนกระบวนการรันเสร็จสิ้น



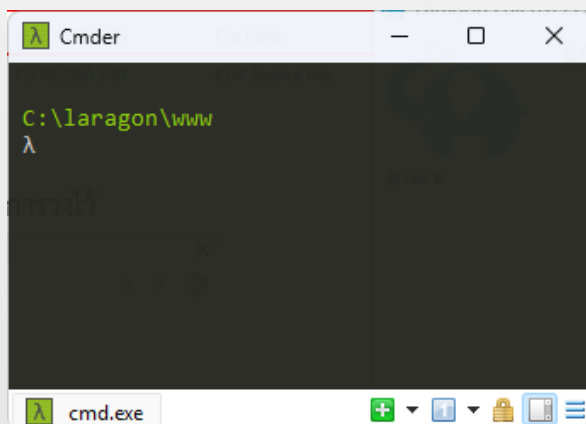
6. หากสร้างโปรเจกต์ได้สำเร็จ จะมีโฟลเดอร์ “MyB68xxxxCRUD” ขึ้นมา



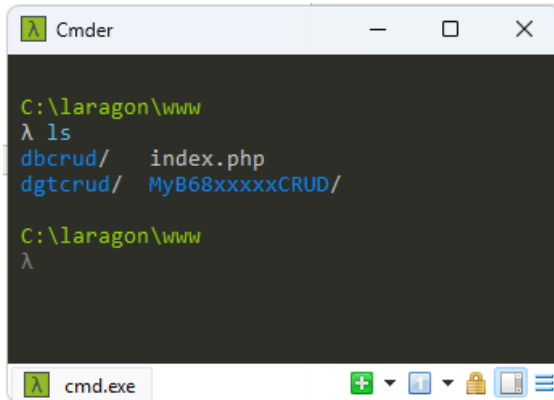
7. จากนั้นให้เปิด Terminal ผ่าน Laragon กดตามที่กรอบได้ทำการวงไว้



จะได้หน้าต่างนี้ขึ้นมา



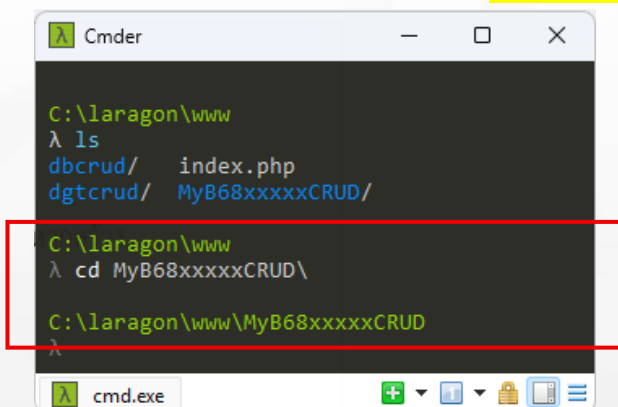
8. ให้ลองใช้คำสั่ง ls (คำสั่งเหมือน linux) แล้วจะได้รายการไฟล์ขึ้นมา จะพบโฟลเดอร์ MyB68xxxxxCRUD/ หากสร้างโปรเจกต์ได้ถูกต้องจะพบโฟลเดอร์ดังกล่าว



```
C:\laragon\www
λ ls
dbcrud/   index.php
dgtcrud/  MyB68xxxxxCRUD/

C:\laragon\www
λ
```

9. หากพบโปรเจกต์แล้วแสดงโฟลเดอร์ ให้ทำการเข้าไปที่โฟลเดอร์นั้น โดยใช้คำสั่ง cd โดยพิมพ์ cd MyB68xxxxxCRUD หากเข้าไปที่โฟลเดอร์ได้ Path จะเปลี่ยนเป็น C:\laragon\www\MyB68xxxxxCRUD

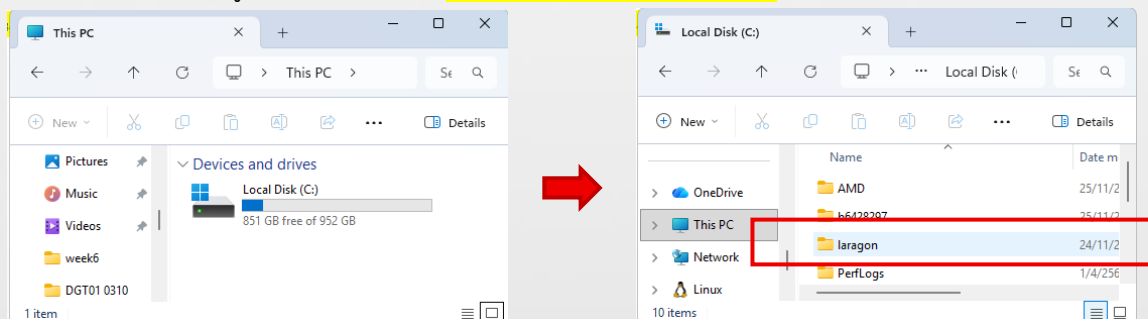


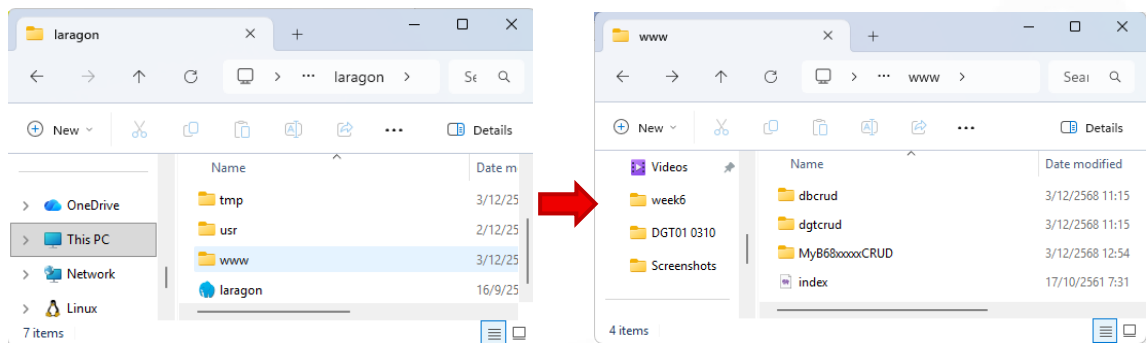
```
C:\laragon\www
λ ls
dbcrud/   index.php
dgtcrud/  MyB68xxxxxCRUD/

C:\laragon\www
λ cd MyB68xxxxxCRUD\

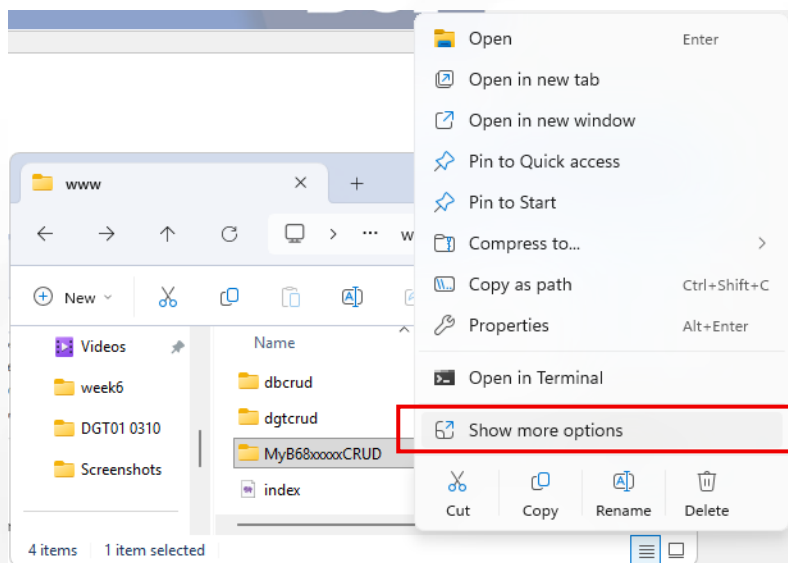
C:\laragon\www\MyB68xxxxxCRUD
λ
```

10. จากนั้นให้เข้าโฟลเดอร์ที่เก็บข้อมูลเว็บในโปรแกรม Laragon ผ่าน File explore แล้วเข้าไปที่ โฟลเดอร์ในเครื่องของนักศึกษา ให้ Path ที่อยู่ไฟล์ใกล้เคียงตามนี้ C:\laragon\www\MyB68xxxxxCRUD

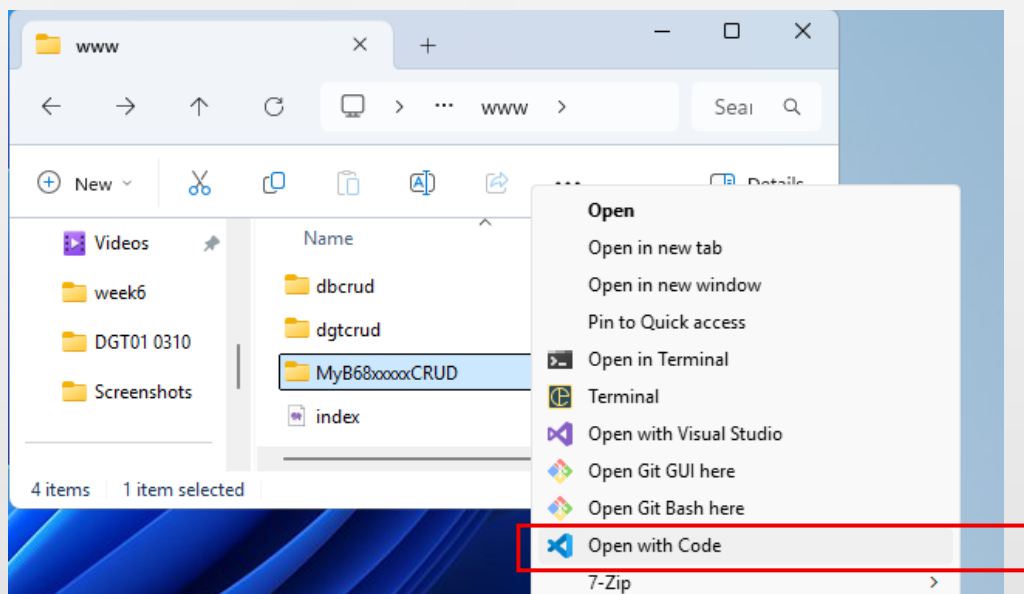




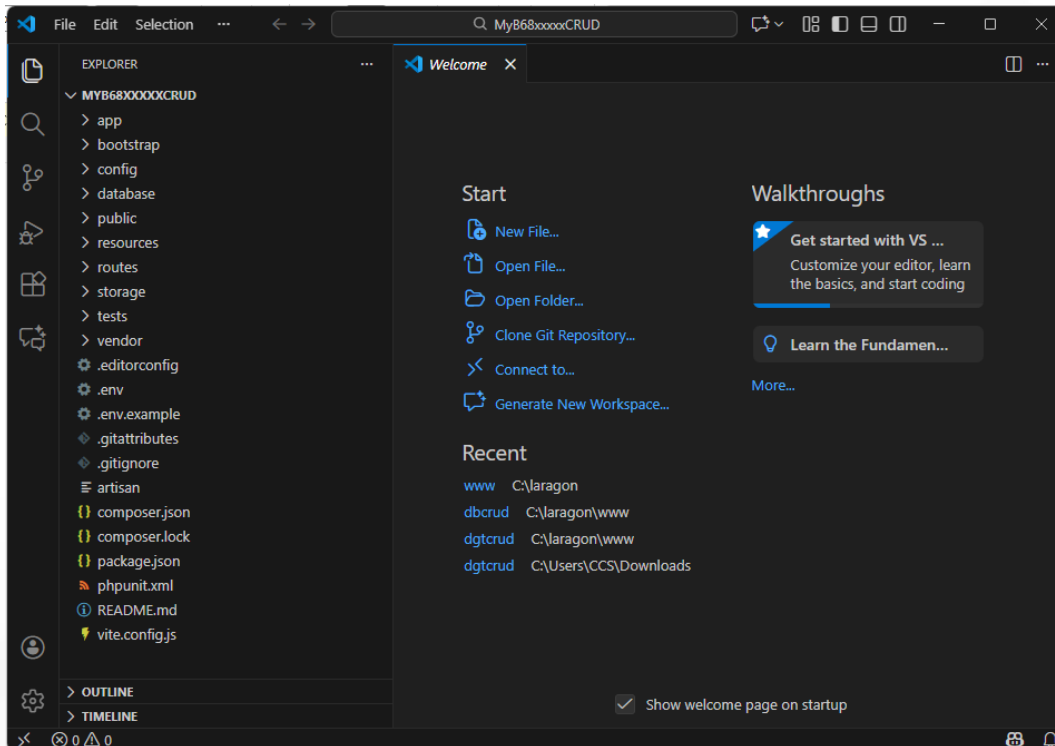
11. จากนั้นคลิกขวาที่โฟลเดอร์ MyB68xxxxCRUD > จากนั้น กด “show more options”



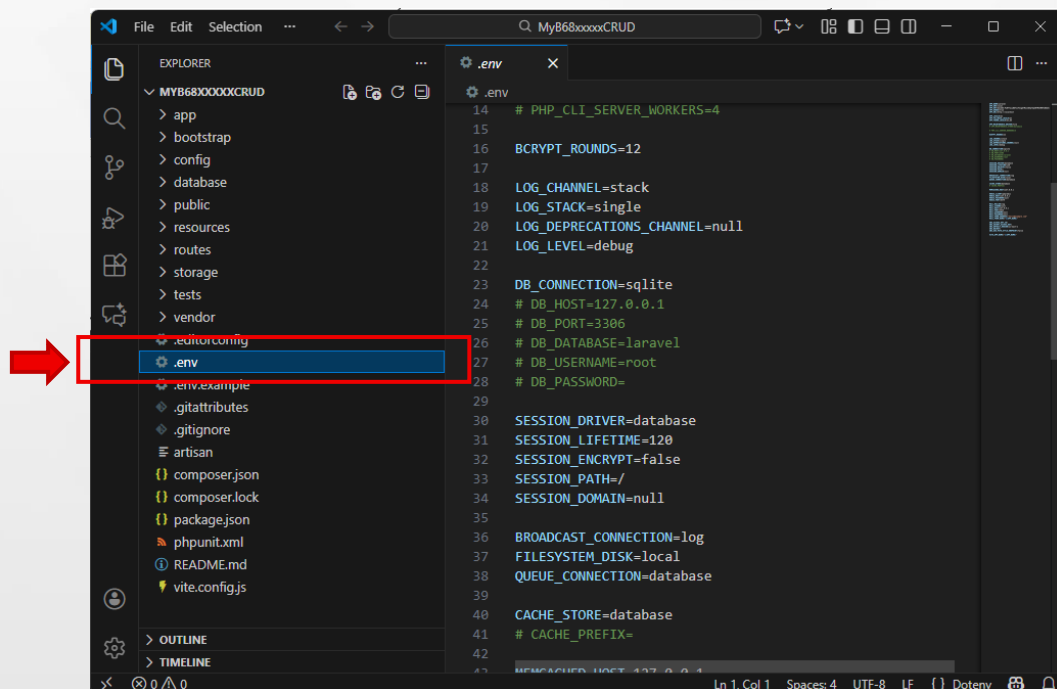
จากนั้นเลือก “Open with Code”



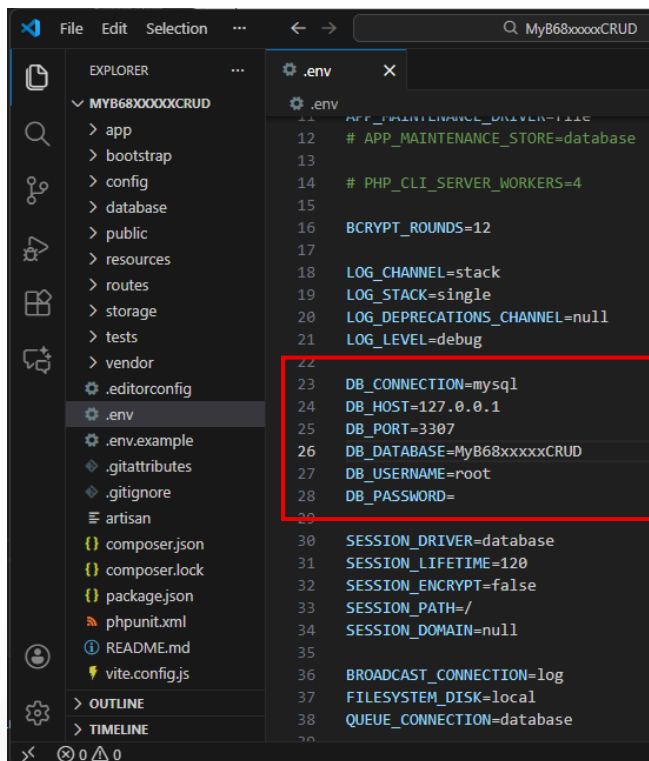
12. จะมีหน้าต่าง vscode ขึ้นมาพร้อมโปรเจกต์ไฟล์โค้ด โฟลเดอร์ และไฟล์ตั้งค่าต่างๆ ตามนี้



13. จากนั้นเปิดไฟล์ .env เพื่อแก้ไขโค้ดในไฟล์ เมื่อเปิดไฟล์แล้วจะมีโค้ดขึ้นมา



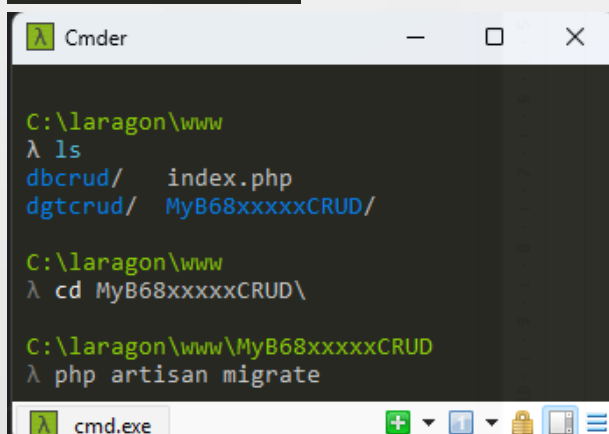
ให้แก้ไขไฟล์ .env แล้วให้หาโค้ดที่เหมือนกับกรอบและลูกศรที่วง จากนั้นแก้ตามรูปต่อไปจากนี้



```
11 APP_MAINTENANCE_DRIVER=file
12 # APP_MAINTENANCE_STORE=database
13
14 # PHP_CLI_SERVER_WORKERS=4
15
16 BCRYPT_ROUNDS=12
17
18 LOG_CHANNEL=stack
19 LOG_STACK=single
20 LOG_DEPRECATIONS_CHANNEL=null
21 LOG_LEVEL=debug
22
23 DB_CONNECTION=mysql
24 DB_HOST=127.0.0.1
25 DB_PORT=3307
26 DB_DATABASE=MyB68xxxxxCRUD
27 DB_USERNAME=root
28 DB_PASSWORD=
29
30 SESSION_DRIVER=database
31 SESSION_LIFETIME=120
32 SESSION_ENCRYPT=false
33 SESSION_PATH=/
34 SESSION_DOMAIN=null
35
36 BROADCAST_CONNECTION=log
37 FILESYSTEM_DISK=local
38 QUEUE_CONNECTION=database
```

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

14. จากนั้นกลับไป Terminal ของ Laragon (Cmder) แล้วพิมพ์คำสั่งต่อไปนี้
- ```
php artisan migrate
```
- จากนั้นกด Enter



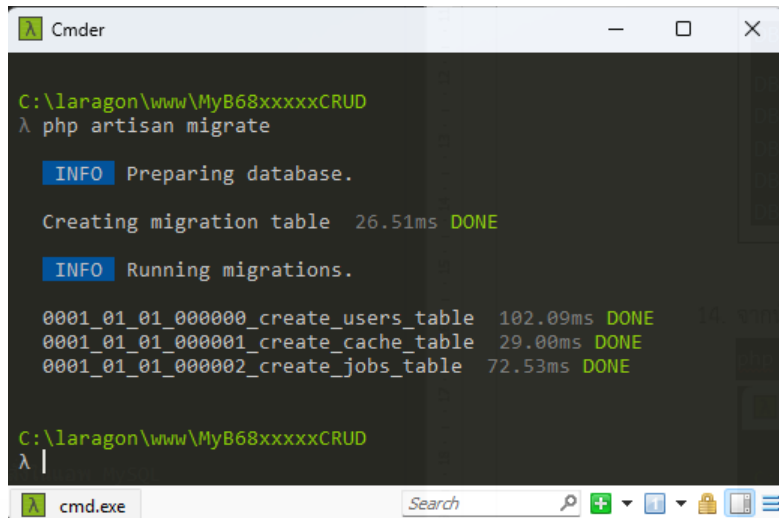
```
C:\laragon\www
λ ls
dbcrud/   index.php
dgtcrud/  MyB68xxxxxCRUD/

C:\laragon\www
λ cd MyB68xxxxxCRUD\

C:\laragon\www\MyB68xxxxxCRUD
λ php artisan migrate
```



15. เมื่อทำการใช้คำสั่งก่อนหน้านี้ หากสามารถรันทำงานได้สำเร็จ จะได้ข้อความเหล่านี้



```
C:\laragon\www\MyB68xxxxxCRUD
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

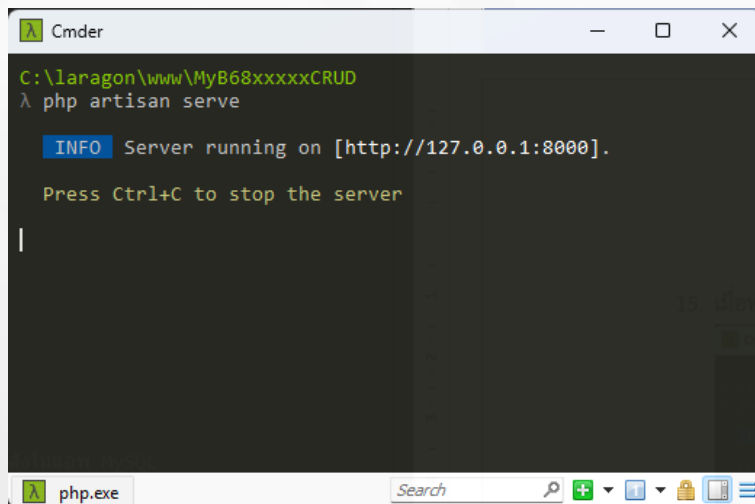
C:\laragon\www\MyB68xxxxxCRUD
λ |
```

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

**php artisan serve**

จากนั้นจะได้ผลลัพธ์ดังรูป จากนั้นให้กดปุ่ม Ctrl ค้างไว้กับนำ Cursor ของเมาส์ไปชี้แล้วกดที่ลิงค์

[http://127.0.0.1:8000]



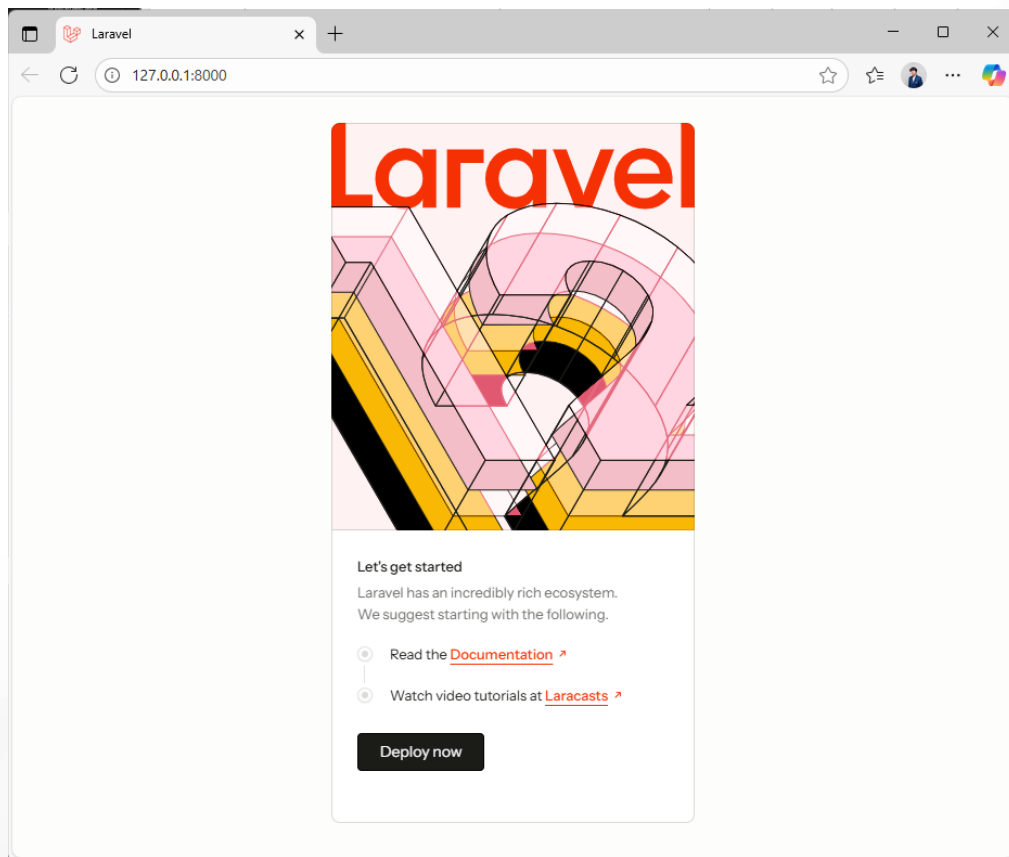
```
C:\laragon\www\MyB68xxxxxCRUD
λ php artisan serve

INFO: Server running on [http://127.0.0.1:8000].

Press Ctrl+C to stop the server

|
```

จากนั้นโปรแกรมเว็บเบราว์เซอร์ปรากฏขึ้นมา



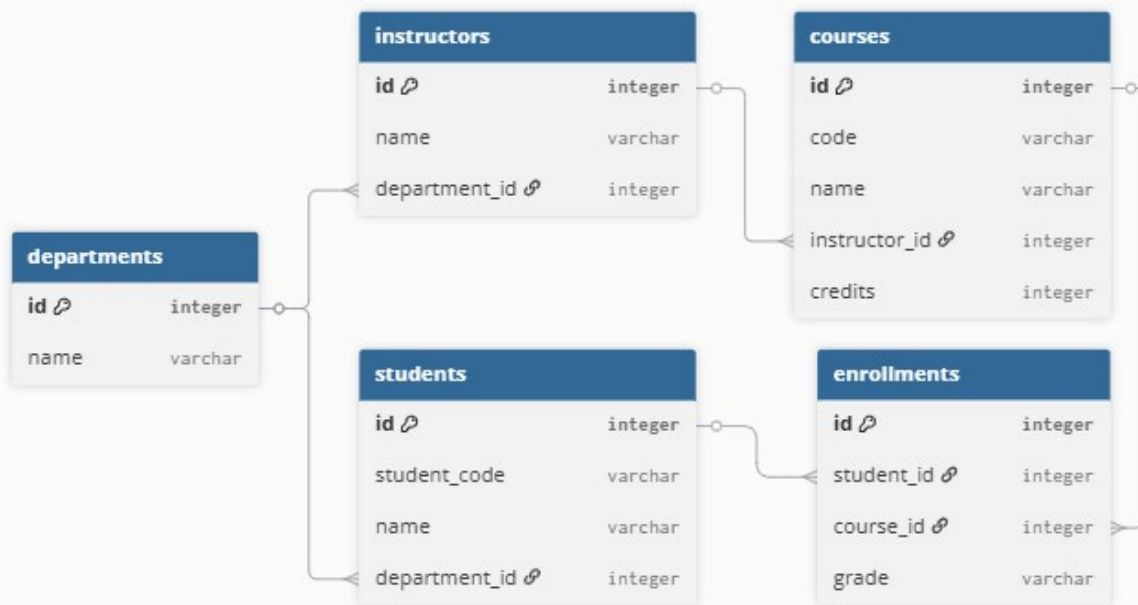
## ส่วนที่ 2: การสร้างฐานข้อมูล ตาราง วิว และการเพิ่มข้อมูล

โปรเจ็คต่อไปนี้เป็น การสร้างโปรเจ็คสำหรับการเรียนวิชา Database development fundamental เพราะจะได้เห็นภาพทั้งการออกแบบ (Design), การจัดการข้อมูลระดับ SQL (Views/Joins), และการนำมาใช้จริงใน Application (Laravel) โดยโปรเจ็คนี้เกี่ยวกับ “ระบบลงทะเบียนเรียนมหาวิทยาลัย (University Course Registration)” โดยเริ่มกระบวนการต่อไปดังนี้

1. การออกแบบฐานข้อมูล (Database Schema) เราจะออกแบบ 4 ตารางหลัก (+1 ตารางสำหรับเชื่อมความสัมพันธ์) เพื่อให้สามารถ Join กันได้ครบทุกตาราง โดยมีตาราง ดังนี้

Entity Relationship (ER):

- 1.1. Departments (คณะ/ภาควิชา): เก็บข้อมูลสังกัด
- 1.2. Students (นักศึกษา): สังกัดภาควิชา (FK)
- 1.3. Instructors (อาจารย์): สังกัดภาควิชา (FK)
- 1.4. Courses (วิชา): สอนโดยอาจารย์ท่านไหน (FK)
- 1.5. Enrollments (การลงทะเบียน): ตาราง Transaction เชื่อมระหว่าง Student และ Course



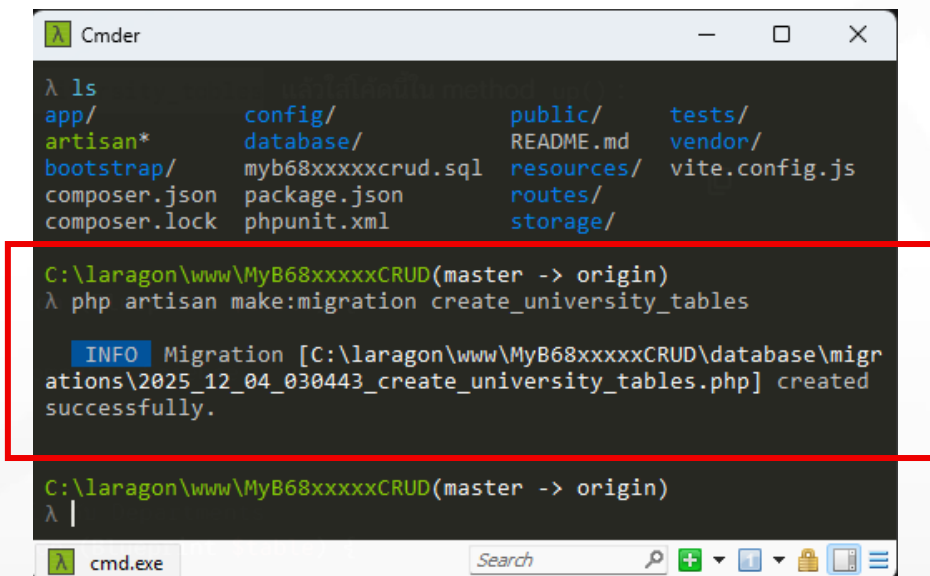
## 2. ขั้นตอนการทำใน Laravel (Migration & Seeding)

### 2.1. สร้างตาราง (Migrations)

ให้ใช้คำสั่งต่อไปนี้ในการสร้างตาราง ใน Terminal Laragon (Cmder)

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

จากนั้นกด Enter แล้วรอกันกว่าจะขึ้นข้อความ “Created successfully”



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

```
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 success message, and a red arrow points to it from the right.

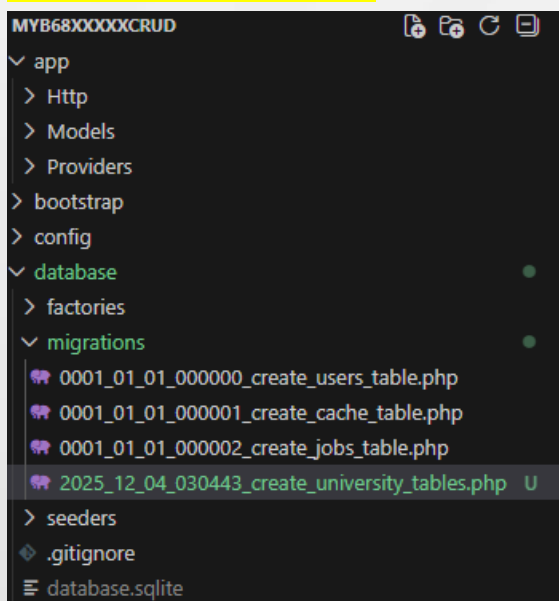
### 2.2. หลังจากสร้างตารางแล้วจะได้ไฟล์ขึ้นมา ให้ใส่โค้ดที่ไฟล์ใหม่ต่อไปนี้ในไฟล์ที่มีชื่อว่า

```
2025_11_18_033331_create_university_tables.php
```

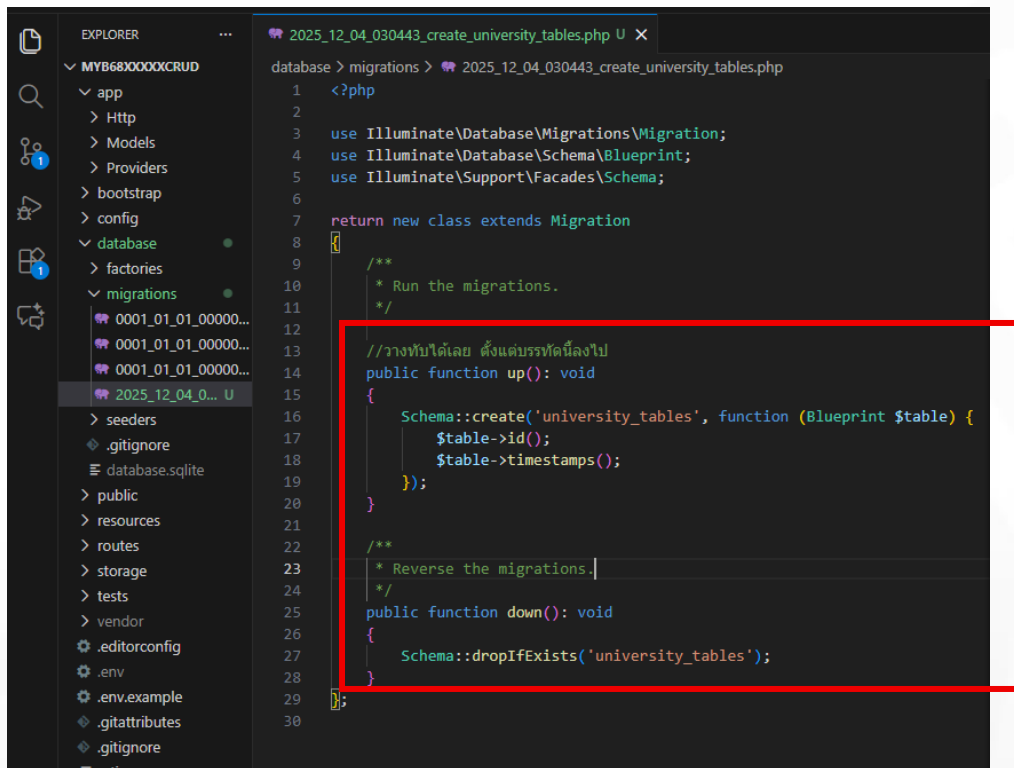
ในโฟลเดอร์ database/migrations

\*\*\*ชื่อไฟล์อาจไม่เหมือนกันกับตัวอย่าง ให้ดูช่วงข้อความชื่อไฟล์ด้านหลังคือ

```
create_university_tables.php
```



จะเห็นว่าให้โค้ด function up() และ down() ขึ้นมา



```
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     //วางทับได้เลย ดึงแบบรวบรัดนี้ลงไป
13     public function up(): void
14     {
15         Schema::create('university_tables', function (Blueprint $table) {
16             $table->id();
17             $table->timestamps();
18         });
19     }
20
21     /**
22      * Reverse the migrations.
23      */
24     public function down(): void
25     {
26         Schema::dropIfExists('university_tables');
27     }
28 };
29
30
```

### 2.3. ให้นำโค้ดต่อไปนี้ ไปวางทับที่ function up() และ down()

```
public function up(): void
{
    // 1. ตาราง Departments (คณะ)
    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');
}
// สิ้นสุดการวางทับ

```

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       //วางทับได้เลย ตั้งแบบรวบรัดลงไป
14       public function up(): void
15       {
16           // 1. ตาราง Departments (คณะ)
17           Schema::create('departments', function (Blueprint $table) {
18               $table->id();
19               $table->string('name');
20               $table->timestamps();
21           });
22
23           // 2. ตาราง Instructors (อาจารย์)
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           });
30
31           // 3. ตาราง Students (นักศึกษา)
32           Schema::create('students', function (Blueprint $table) {
33               $table->id();
34               $table->string('student_code');
35               $table->string('name');
36               $table->foreignId('department_id')->constrained('departments');
37               $table->timestamps();
38           });
39
40           // 4. ตาราง Courses (วิชา)
41           Schema::create('courses', function (Blueprint $table) {
42               $table->id();
43               $table->string('code');
44               $table->string('name');
45               $table->foreignId('instructor_id')->constrained('instructors');
46               $table->integer('credits');
47               $table->timestamps();
48           });
49
50           // 5. ตาราง Enrollments (การลงทะเบียน)
51           Schema::create('enrollments', function (Blueprint $table) {
52               $table->id();
53               $table->foreignId('student_id')->constrained('students')->onDelete('cascade');
54               $table->foreignId('course_id')->constrained('courses')->onDelete('cascade');
55               $table->string('grade')->nullable();
56               $table->timestamps();
57           });
58       }
59
60       /**
61       * Reverse the migrations.
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       // สิ้นสุดการวางทับ
72   };
73
```

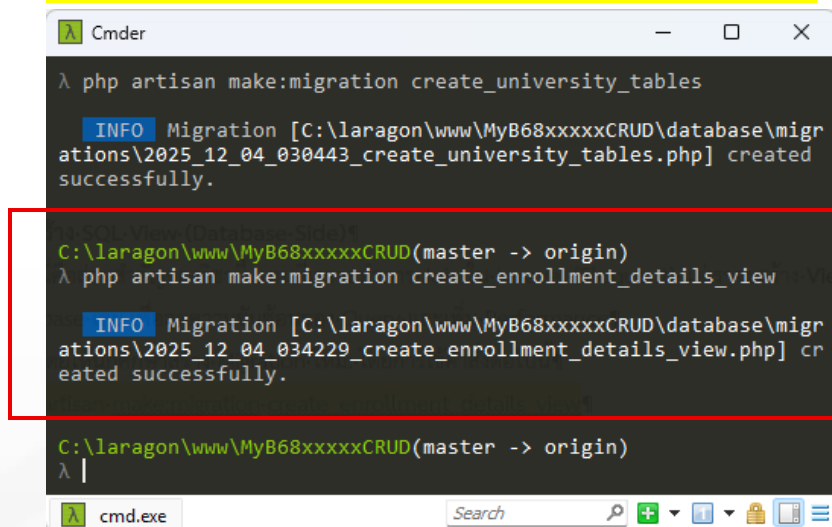


### 3. การสร้าง SQL View (Database Side)

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



```
λ php artisan make:migration create_university_tables

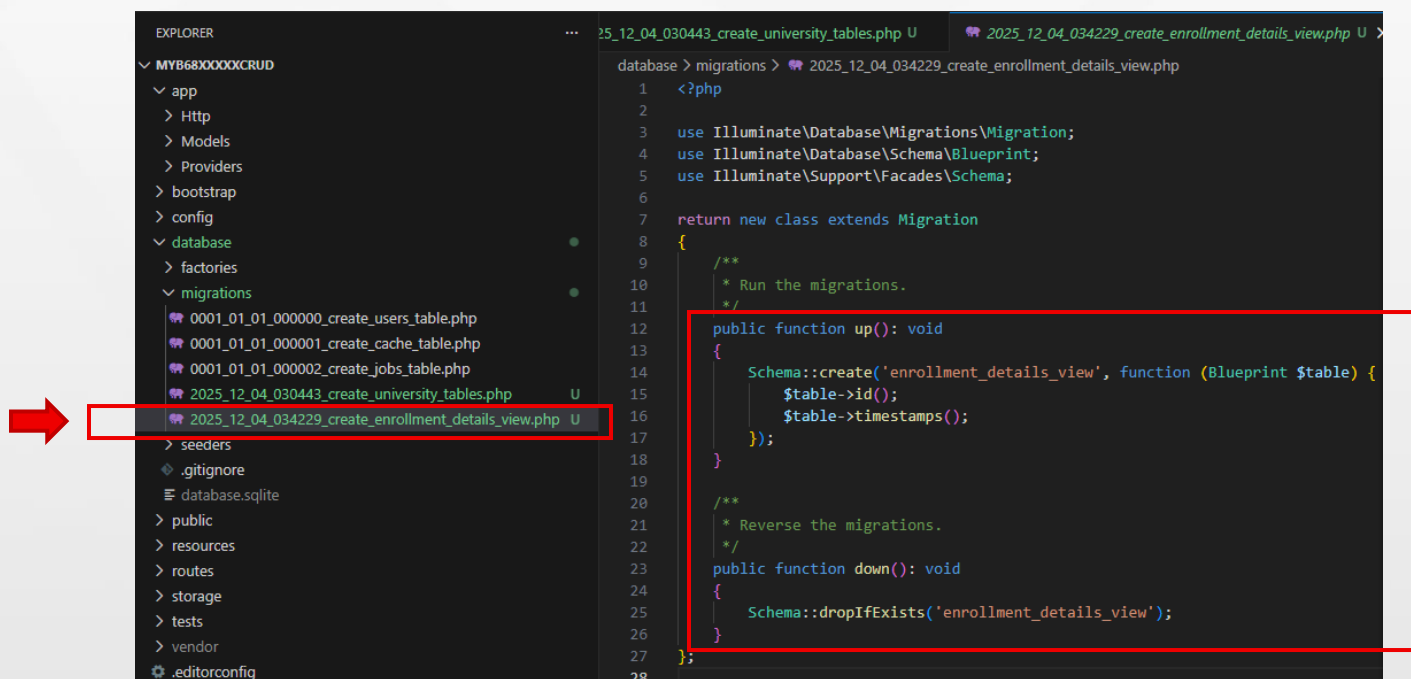
[INFO] Migration [C:\laragon\www\MyB68xxxxxCRUD\database\migrations\2025_12_04_034229_create_university_tables.php] created successfully.

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

[INFO] Migration [C:\laragon\www\MyB68xxxxxCRUD\database\migrations\2025_12_04_034229_create_enrollment_details_view.php] created successfully.

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

จะได้ไฟล์ขึ้นมา และโค้ดดังกล่าว



```
EXPLORER
MYB68XXXXXCRUD
├── app
│   ├── Http
│   ├── Models
│   ├── Providers
│   ├── 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_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

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     public function up(): void
13     {
14         Schema::create('enrollment_details_view', function (Blueprint $table) {
15             $table->id();
16             $table->timestamps();
17         });
18     }
19
20     /**
21      * Reverse the migrations.
22      */
23     public function down(): void
24     {
25         Schema::dropIfExists('enrollment_details_view');
26     }
27 }
28
```

\*\*\*ชื่อไฟล์อาจไม่เหมือนกันกับตัวอย่าง ให้ดูช่วงข้อความชื่อไฟล์ด้านหลัง



### 3.2. นำโค้ดต่อไปนี ไปวางทับที่ function up() และ down()

```
// วางทับได้เลย ตั้งแต่บรรทัดนี้ลงไป

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");
}

// สิ้นสุดการวางทับ
```

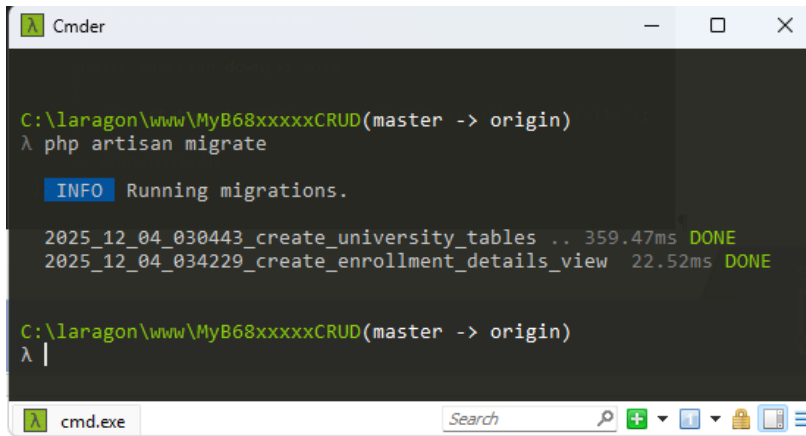
ตัวอย่างการวางทับโค้ดของไฟล์ 2025\_12\_04\_034229\_create\_enrollment\_details\_view.php

\*\*\*ระวางฝัง ระวางปักกา

```
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       // วางทับได้เลย 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       // สิ้นสุดการวางทับ
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)

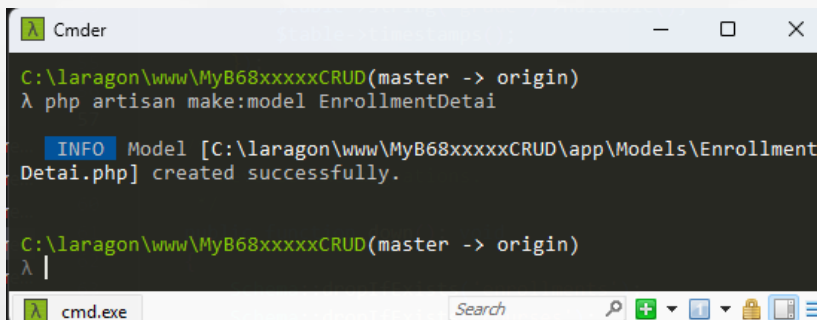
### 3.4. การเชื่อมต่อและทำ CRUD ใน Laravel

Model Setup

สร้าง Model สำหรับ View ด้วย (ทำเหมือนตารางปกติ แต่ปิด timestamp) ใช้คำสั่ง

php artisan make:model EnrollmentDetail

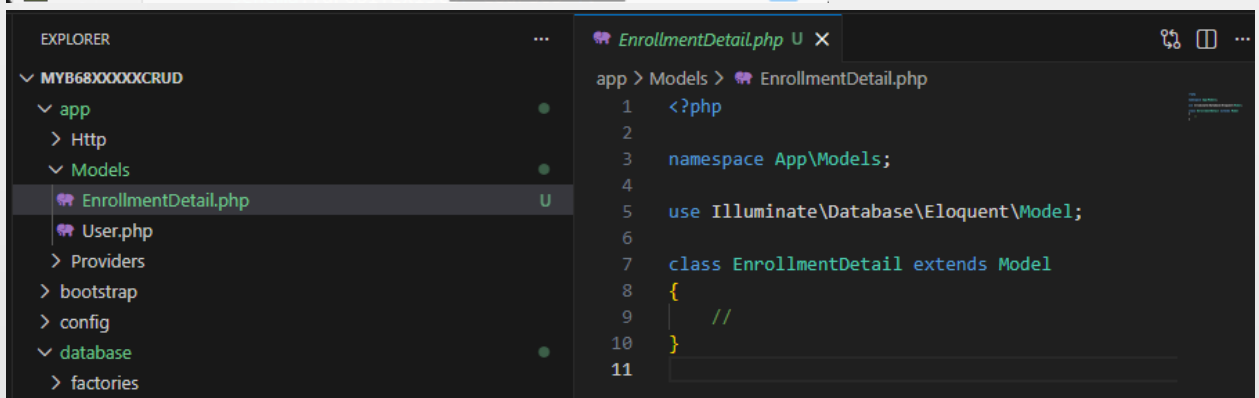
เมื่อรันคำสั่งเสร็จแล้วจะได้ไฟล์ EnrollmentDetail.php



```
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 |
```



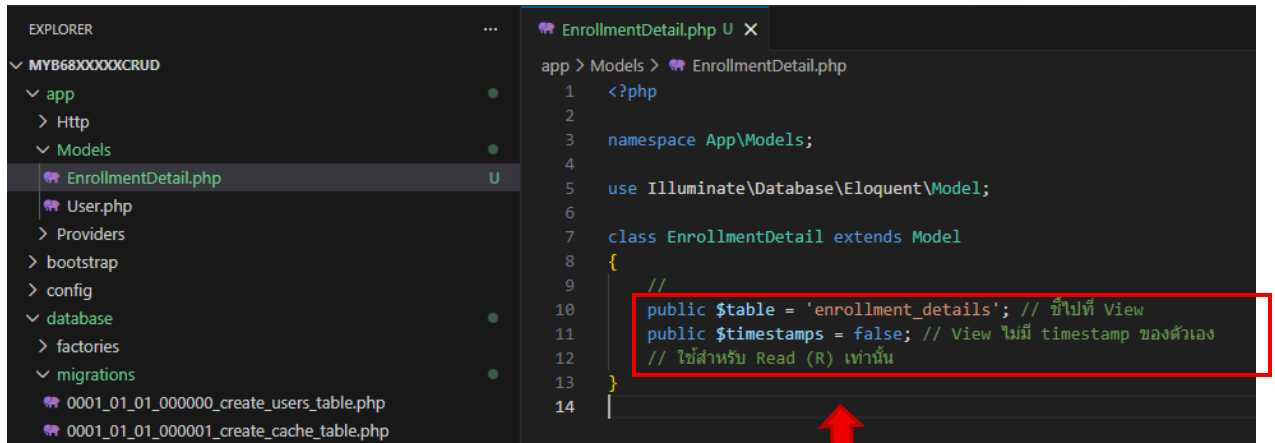
EXPLORER

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

EnrollmentDetail.php

```
1 <?php
2
3 namespace App\Models;
4
5 use Illuminate\Database\Eloquent\Model;
6
7 class EnrollmentDetail extends Model
8 {
9     //
10 }
11
```

### 3.5. เปิดไฟล์ EnrollmentDetail.php ในโฟลเดอร์ app/Models แล้วเพิ่มโค้ดตามกรอบ



```
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'; // ชื่อที่ View
11    public $timestamps = false; // View ไม่มี timestamp ของตัวเอง
12    // ใช้สำหรับ Read (R) เท่านั้น
13 }
14
```

โค้ด

```
public $table = 'enrollment_details'; // ชื่อที่ 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

\*\*\*ผลการรันอยู่ทางด้านขวานี้



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.

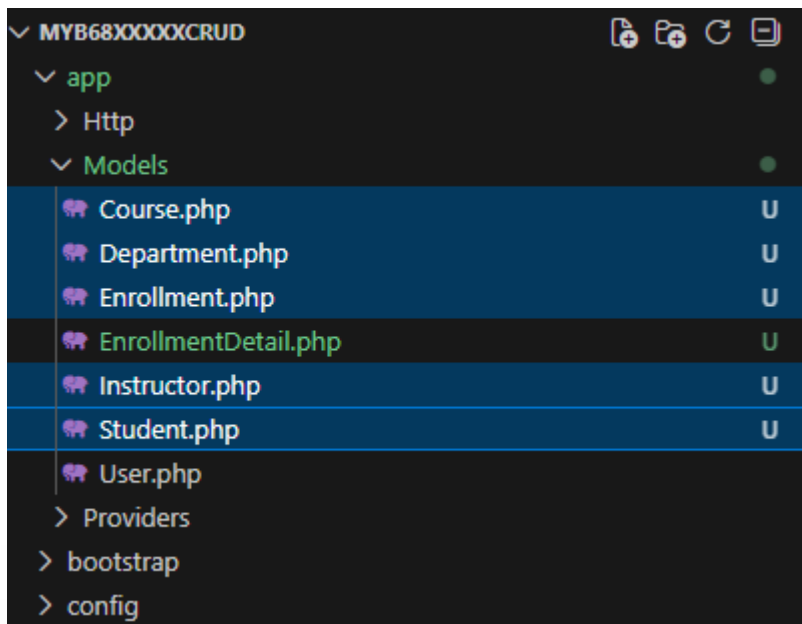
C:\laragon\www\MyB68xxxxxCRUD(master -> origin)
λ php artisan make:model Instructor
[INFO] Model [C:\laragon\www\MyB68xxxxxCRUD\app\Models\Instructor.php] created successfully.

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.

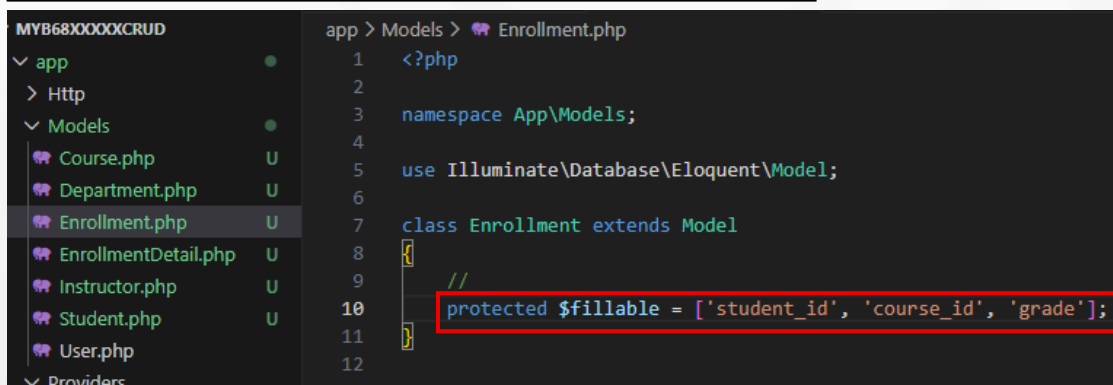
C:\laragon\www\MyB68xxxxxCRUD(master -> origin)
λ php artisan make:model Enrollment
[INFO] Model [C:\laragon\www\MyB68xxxxxCRUD\app\Models\Enrollment.php] created successfully.

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

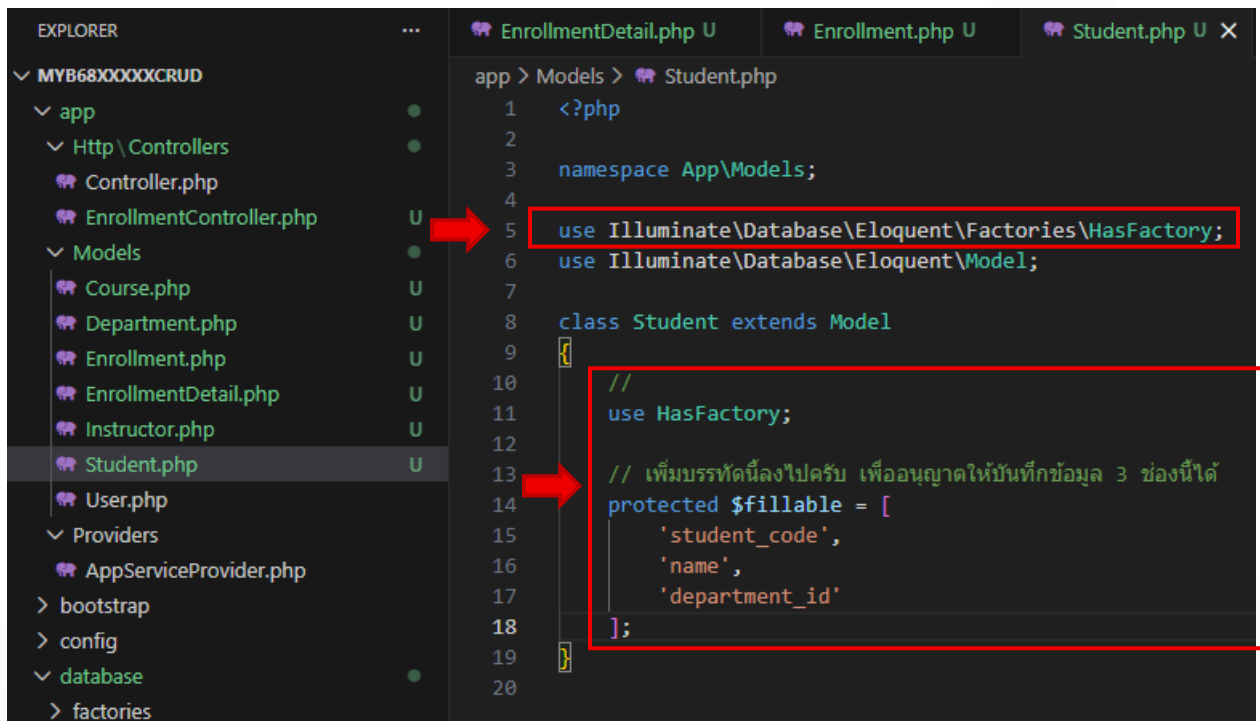


3.7. จากนั้นเปิดไฟล์ที่มีชื่อว่า “Enrollment.php” แล้วเพิ่มโค้ดต่อไปนี้

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



3.8. จากนั้นให้เปิดไฟล์ Student.php ที่อยู่ในโฟลเดอร์ app/Models/ แล้วให้เพิ่ม โค้ดและตัวแปรตามรูปตัวอย่างและโค้ดที่ล้อมกรอบแดงไว้ ดังนี้



```
<?php

namespace App\Models;

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

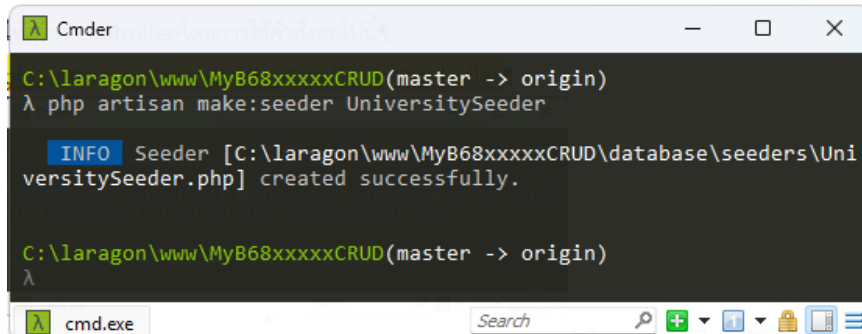
class Student extends Model
{
    //
    use HasFactory;

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

### 3.9. สร้างข้อมูลจำลอง (Seeding)

ฟีเจอร์นี้ทำให้นักศึกษามีข้อมูลเล่นทันที โดยไม่ต้องนั่งกรอกเอง สร้าง Seeder โดยพิมพ์คำสั่ง:

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

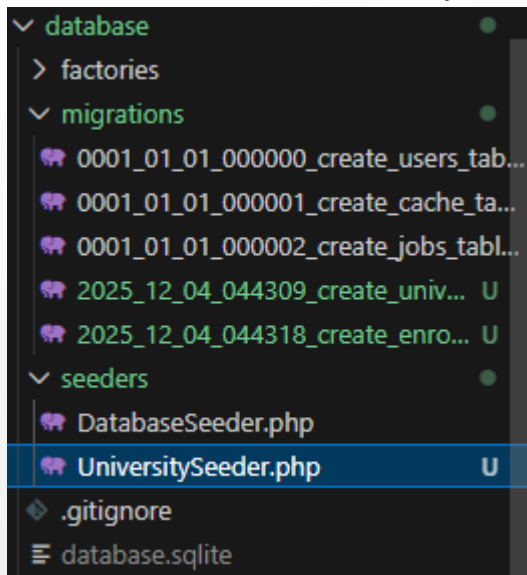


```
C:\laragon\www\MyB68xxxxxCRUD(master -> origin)
λ php artisan make:seeder UniversitySeeder

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

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

จะได้ไฟล์ UniversitySeeder.php ที่อยู่ภายใต้โฟลเดอร์ database/migrations/seeders



จากนั้นให้เพิ่มโค้ดตามรูปลงไป

\*\*\*รูปและโค้ดอยู่ในหน้าถัดไป

```
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 รักเรียน', '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
```

\*\*\*โค้ดอยู่ที่หน้าถัดไป



```

<?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. สร้างคณะ
        $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.วิภา รักเรียน', 'department_id' => $it->id]);

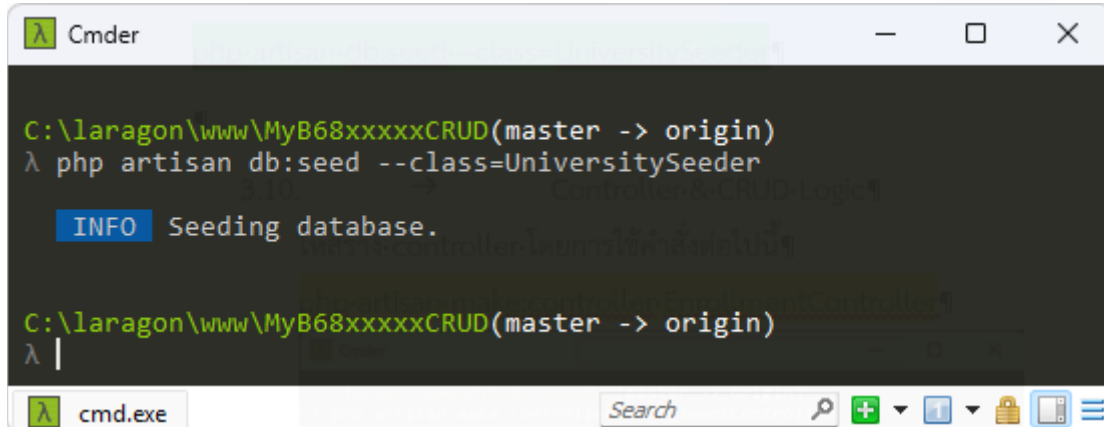
        // 3. สร้างนักศึกษา
        Student::create(['student_code' => '66001', 'name' => 'นาย ก.', 'department_id' =>
$cs->id]);
        Student::create(['student_code' => '66002', 'name' => 'นางสาว ข.', 'department_id' =>
$it->id]);

        // 4. สร้างวิชา
        Course::create(['code' => 'CS101', 'name' => 'เขียนโปรแกรมเบื้องต้น', '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\MyB68xxxxxCRUD(master -> origin)
λ php artisan db:seed --class=UniversitySeeder

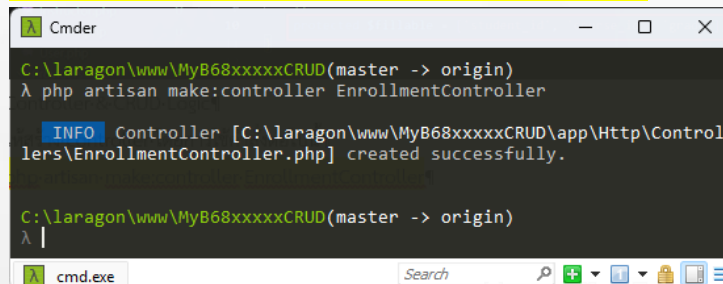
INFO Seeding database.

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

### 3.10. Controller & CRUD Logic

ให้สร้าง controller โดยการใช้คำสั่งต่อไปนี้

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

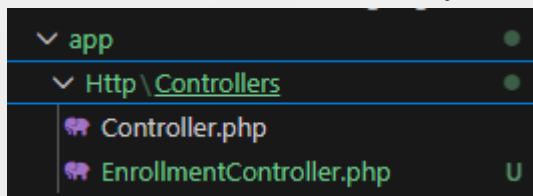


```
C:\laragon\www\MyB68xxxxxCRUD(master -> origin)
λ php artisan make:controller EnrollmentController

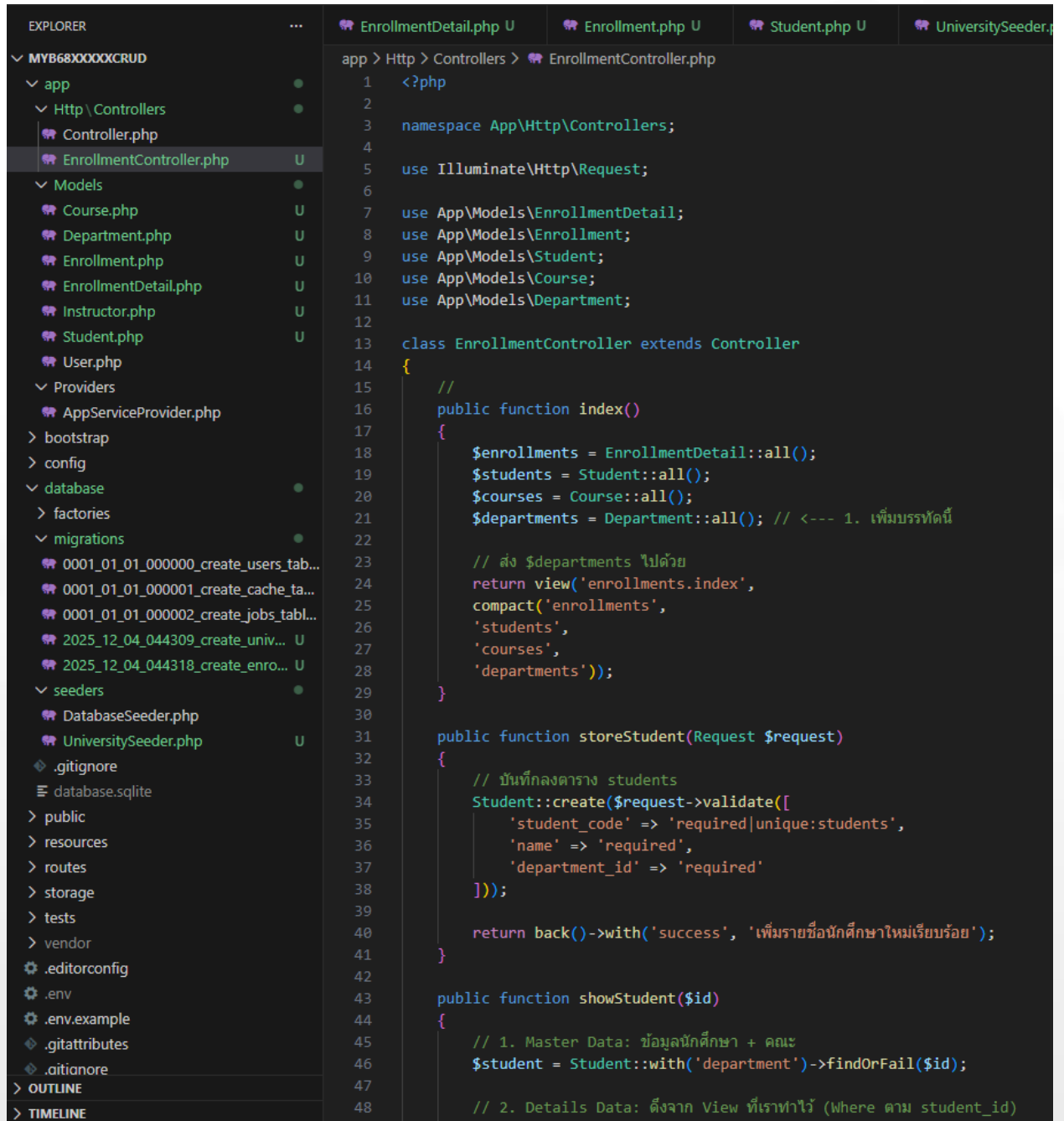
INFO Controller [C:\laragon\www\MyB68xxxxxCRUD\app\Http\Controllers\EnrollmentController.php] created successfully.

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

เราจะได้ไฟล์ของ controller ขึ้นมา ดังรูปภาพ



### 3.11. ให้เพิ่มโค้ดต่อไปนี้ตามตัวอย่างรูปภาพ ที่ไฟล์ 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 $request)
32     {
33         // บันทึกลงตาราง students
34         Student::create($request->validate([
35             'student_code' => 'required|unique:students',
36             'name' => 'required',
37             'department_id' => 'required'
38         ]));
39
40         return back()->with('success', 'เพิ่มรายชื่อนักศึกษาใหม่เรียบร้อยแล้ว');
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\_000000\_create\_users\_tab...

0001\_01\_01\_000001\_create\_cache\_ta...

0001\_01\_01\_000002\_create\_jobs\_tabl...

2025\_12\_04\_044309\_create\_univ...

2025\_12\_04\_044318\_create\_enro...

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

14

{

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

67

68

69

70

71

72

73

74

75

76

77

78

79

80

81

82

83

84

85

86

87

88

89

public function showStudent(\$id)

{

// 1. Master Data: ข้อมูลนักศึกษา + คณะ

\$student = Student::with('department')->findOrFail(\$id);

// 2. Details Data: ดึงจาก View ที่เราทำไว้ (Where ตาม student\_id)

\$history = EnrollmentDetail::where('student\_id', \$id)->get();

// 3. คำนวณหน่วยกิตรวม (ลูกเล่นเพิ่มเติม)

\$totalCredits = \$history->sum(function(\$row) {

// (ต้องดึง credit จาก Course จริง หรือจะ Join มาใน View ก็ได้)

// เพื่อความง่ายในตัวอย่างนี้ ผมขอข้ามการดึง Credit ไปก่อน หรือ

// ถ้าใน View ยังไม่มี credit ให้กลับไปแก้ View เพิ่ม column c.credits

return 0;

});

return view('enrollments.student\_detail', compact('student', 'his

}

public function store(Request \$request)

{

// บันทึกข้อมูลลง Table จริง

Enrollment::create(\$request->validate([

'student\_id' => 'required',

'course\_id' => 'required',

'grade' => 'nullable'

]));

return back()->with('success', 'ลงทะเบียนเรียบร้อยแล้ว');

}

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

{

// อัปเดตเกรด

\$enrollment = Enrollment::find(\$id);

\$enrollment->update(['grade' => \$request->grade]);

return back()->with('success', 'อัปเดตเกรดเรียบร้อยแล้ว');

}

public function destroy(\$id)

{

// ลบข้อมูล

Enrollment::destroy(\$id);

return back()->with('success', 'ถอนรายวิชาเรียบร้อยแล้ว');

}

// เพิ่มฟังก์ชันนี้ต่อท้ายสุดใน Controller

public function report(Request \$request)

มีต่อในหน้าถัดไป

```
14 {
88 // เพิ่มฟังก์ชันนี้คือท้ายสุดใน Controller
89 public function report(Request $request)
90 {
91     // 1. ดึงรายชื่อวิชาทั้งหมด เพื่อเอาไปใส่ใน Dropdown ตัวเลือก
92     $courses = Course::all();
93
94     // 2. ตัวแปรเก็บผลลัพธ์ (เริ่มต้นเป็นค่าว่าง)
95     $results = collect();
96     $selectedCourse = null;
97
98     // 3. ถ้ามีการเลือกวิชามา (กดปุ่มค้นหา)
99     if ($request->has('course_id')) {
100         $courseId = $request->course_id;
101
102         // ดึงข้อมูลจาก View โดย Filter ตาม Course ID
103         $results = EnrollmentDetail::where('course_id', $courseId)
104             ->orderBy('student_code') // เรียงตามรหัสนักศึกษา
105             ->get();
106
107         // ดึงข้อมูลวิชาที่เลือกมาโชว์หัวกระดาษ
108         $selectedCourse = Course::find($courseId);
109     }
110
111     return view('enrollments.report',
112         compact('courses', 'results', 'selectedCourse'));
113 }
114 }
115 }
```

\*\*\*ลิงค์คัดลอกโค้ด

<https://raw.githubusercontent.com/pyrnmsutdgt/MyB68xxxxxCRUD/refs/heads/CR/app/Http/Controllers/EnrollmentController.php>

โค้ดชุดนี้เป็น **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 เลือกคณะตอนสมัครนักศึกษาใหม่)
- **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()) พร้อมข้อความแจ้งเตือนว่าเพิ่มเรียบร้อยแล้ว
- ## 3. public function showStudent(\$id)
- หน้าที่: แสดงหน้ารายละเอียดของนักศึกษาคนนั้นๆ (Profile & History)
- **การทำงาน:** แบ่งเป็น 3 ส่วน
    1. **Master Data:** ค้นหานักศึกษาตาม \$id และดึงข้อมูลแผนก (department) มาด้วย หากไม่เจอจะแจ้ง Error 404 (findOrFail)
    2. **Details Data:** ดึงประวัติการลงทะเบียน (EnrollmentDetail) เฉพาะของนักศึกษาคนนี้
    3. **Calculation:** โค้ดส่วนนี้เตรียมไว้คำนวณหน่วยกิตรวม (sum) แต่ในตัวอย่างนี้ใส่ return 0 ไว้เป็นค่าเริ่มต้น (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 พร้อมแจ้งเตือนว่าถอนวิชาเรียบร้อยแล้ว

#### 7. public function report(Request \$request)

หน้าที่: แสดงรายงานรายชื่อนักศึกษาในแต่ละวิชา (Class Roster)

- **การทำงาน:**
  1. **เตรียมข้อมูล:** ดึงรายชื่อวิชาทั้งหมด (Course::all()) เพื่อไปทำตัวเลือก Dropdown ให้ผู้ใช้กดค้นหา
  2. **ตรวจสอบการค้นหา:** เช็คว่าการส่งค่า 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);

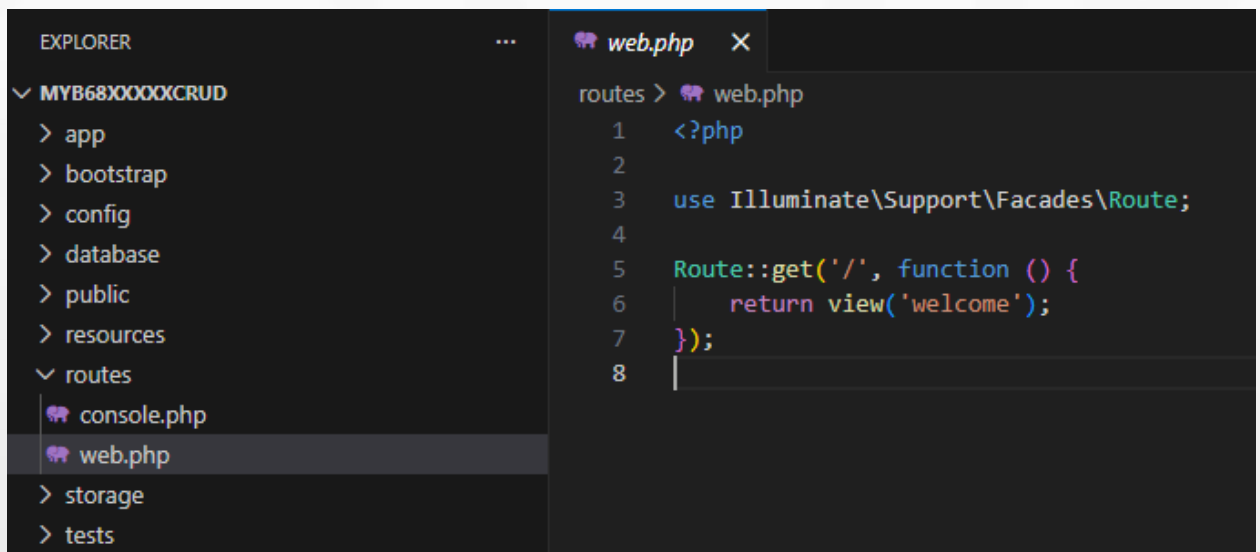
Route::get('/', [EnrollmentController::class, 'index']);

Route::post('/students', [EnrollmentController::class, 'storeStudent'])->name('students.store');

//ขอเพิ่ม route details

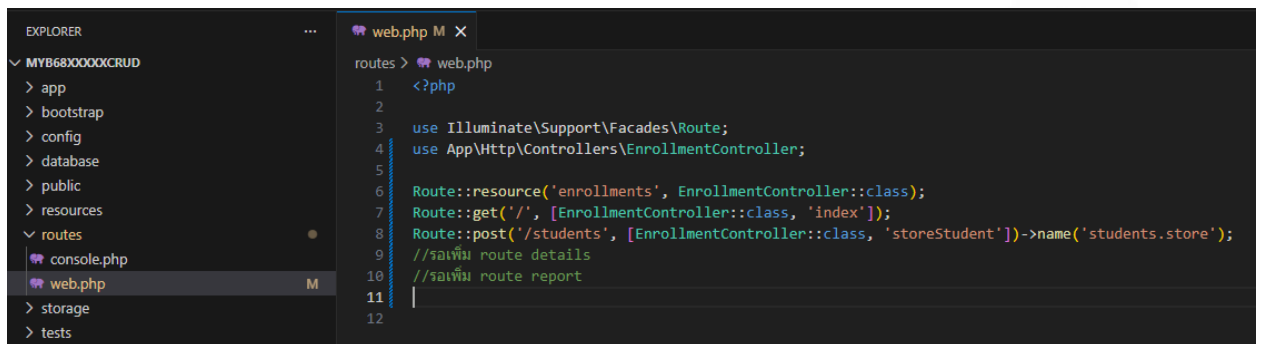
//ขอเพิ่ม route report
```

ก่อนเพิ่ม



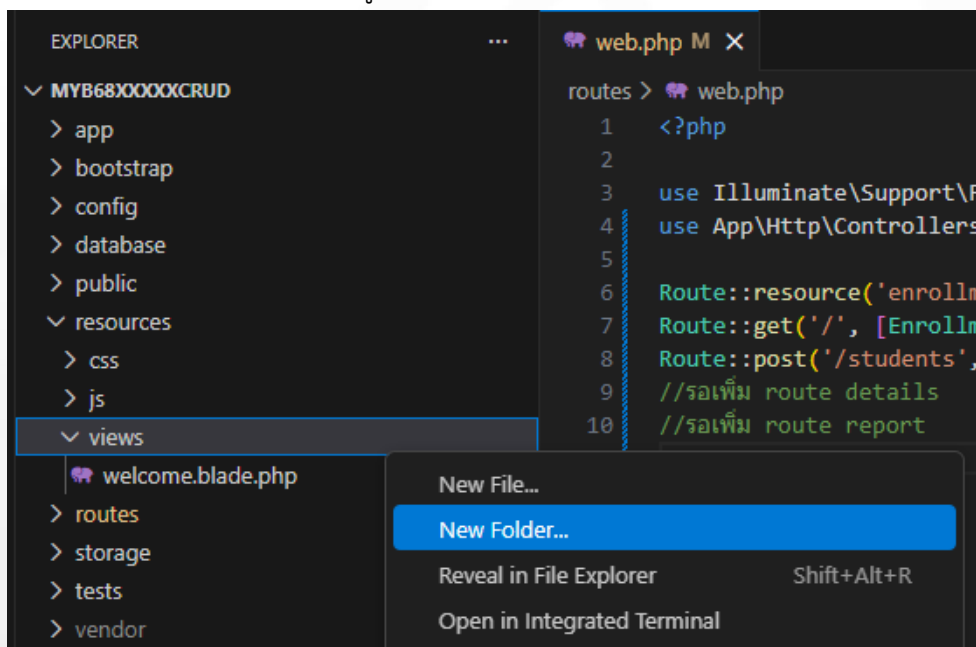
หลังเพิ่ม



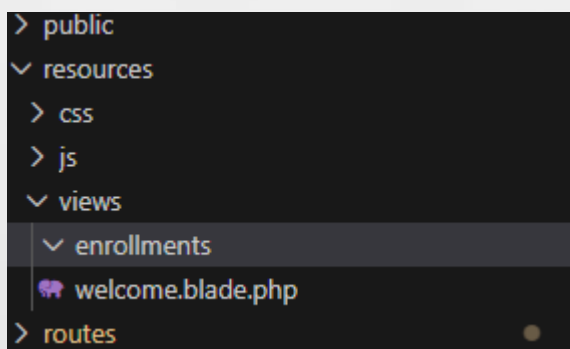


```
1 <?php
2
3 use Illuminate\Support\Facades\Route;
4 use App\Http\Controllers\EnrollmentController;
5
6 Route::resource('enrollments', EnrollmentController::class);
7 Route::get('/', [EnrollmentController::class, 'index']);
8 Route::post('/students', [EnrollmentController::class, 'storeStudent'])->name('students.store');
9 //รอเพิ่ม route details
10 //รอเพิ่ม route report
11
12
```

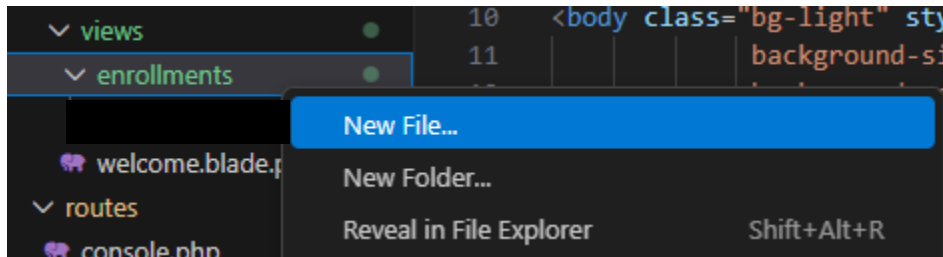
3.13. จากนั้นให้เริ่มสร้างหน้าเว็บ โดยให้สร้างโฟลเดอร์ที่มีชื่อว่า “enrollments” โดยการคลิกขวาที่โฟลเดอร์ views แล้วพิมพ์ชื่อดังรูป



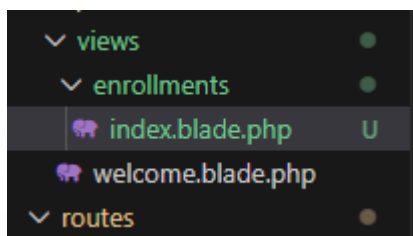
เมื่อสร้างโฟลเดอร์แล้วจะได้ดังนี้



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



จะได้ดังนี้



จากนั้นให้เพิ่มโค้ดต่างในไฟล์ index.blade.php ให้ครบถ้วนตามรูปต่อไปนี้

\*\*\*รูปโค้ดอยู่ในหน้าถัดไป

resources &gt; views &gt; enrollments &gt; 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://cdn.jsdelivr.net/npm/@fortawesome/fontawesome-free@6.0.0/css/all.min.css">
9  </head>
10 <body class="bg-light" style="background-image: url('{{ asset('images/bg.png') }}');
11     background-size: cover;
12     background-position: center;
13     background-repeat: no-repeat;
14     background-attachment: fixed;
15     min-height: 100vh;">
16
17     <div class="container mt-5">
18         <div class="row mb-4">
19             <div class="col-md-8">
20                 <!-- ข้อความหัวข้อ -->
21                 <h2 class="text-primary text-muted"><i class="fas fa-university"></i> ระบบลงทะเบียนเรียน</h2>
22                 <p class="text-muted">ตัวอย่างการใช้ SQL View ร่วมกับ Laravel</p>
23
24                 <!-- ปุ่มรายงาน ให้เพิ่มอาทิยัดไป -->
25
26                 <!-- สิ้นสุดปุ่มรายงาน -->
27
28             </div>
29             <div class="col-md-4 text-end">
30                 <!-- ปุ่มเปิดโมดัลเพิ่มข้อมูล -->
31                 <button type="button" class="btn btn-success" data-bs-toggle="modal" data-bs-target="#createModal">
32                     <i class="fas fa-plus"></i> ลงทะเบียนเพิ่ม
33                 </button>
34                 <!-- ปุ่มเปิดโมดัลเพิ่มนักศึกษาใหม่ -->
35                 <button type="button" class="btn btn-warning ms-2" data-bs-toggle="modal" data-bs-target="#createStudentModal">
36                     <i class="fas fa-user-plus"></i> เพิ่มนักศึกษาใหม่
37                 </button>
38             </div>
39         </div>
40
41         <!-- แสดงข้อความสถานะ -->
42         @if(session('success'))
43             <div class="alert alert-success">{{ session('success') }}</div>
44         @endif
45
46         <!-- ตารางแสดงข้อมูลการลงทะเบียน -->
47         <div class="card shadow-sm">
48             <div class="card-body">
49                 <table class="table table-hover align-middle">
50                     <thead class="table-dark">
51                         <tr>
52                             <th>รหัสนักศึกษา</th>
53                             <th>ชื่อนักศึกษา</th>
54                             <th>คณะ</th>
55                             <th>วิชา</th>
56                             <th>อาจารย์ผู้สอน</th>
57                             <th>เกรด</th>
58                             <th class="text-center">จัดการ</th>
59                         </tr>
60                     </thead>
61                     <tbody>
62                         <!-- รายการลงทะเบียน -->
63                         @foreach($enrollments as $row)
64                             <tr>
65                                 <!-- ID -->
66                                 <td><span class="badge bg-secondary">{{ $row->student_code }}</span></td>
67                                 <!-- ลิงก์ไปยังหน้ารายละเอียดนักศึกษา -->
68                                 <td>
69                                     <!-- ชื่อศึกษาไปหน้า details ได้ ในอาทิยัดหน้า -->
70                                     <a class="text-decoration-none fw-bold">
71   {{ $row->student_name }}
72                                     </a>
73                                 </td>
74                                 <!-- คณะ -->
75                                 <td><small class="text-muted">{{ $row->department_name }}</small></td>
76                                 <!-- ข้อมูลวิชา -->
77                                 <td>
78                                     <strong>{{ $row->course_code }}</strong><br>
79                                     {{ $row->course_name }}
80                                 </td>

```

```

80     </td>
81     <!-- ชื่ออาจารย์ผู้สอน -->|
82     <td>{{ $row->instructor_name }}</td>
83     <!-- เกรด -->
84     <td>
85         @if($row->grade)
86             <span class="badge bg-primary">{{ $row->grade }}</span>
87         @else
88             <span class="badge bg-warning text-dark">รอเกรด</span>
89         @endif
90     </td>
91     <!-- ปุ่มจัดการ -->
92     <td class="text-center">
93         <!-- ปุ่มแก้ไขเกรด ให้เพิ่มอาทิตย์ถัดไป -->
94
95         <!-- สิ้นสุดปุ่มแก้ไขเกรด -->
96         <!-- ปุ่มลบการลงทะเบียน ให้เพิ่มอาทิตย์ถัดไป -->
97
98         <!-- สิ้นสุดปุ่มลบการลงทะเบียน -->
99     </td>
100     <!-- สิ้นสุดปุ่มจัดการ -->
101 </tr>
102
103 <!-- โมดัลแก้ไขเกรด ให้เพิ่มอาทิตย์ถัดไป -->
104
105 <!-- สิ้นสุดโมดัลแก้ไขเกรด -->
106 @endforeach
107 </tbody>
108 </table>
109 </div>
110 </div>
111 </div>
112
113 <!-- โมดัลเพิ่มการลงทะเบียนใหม่ -->
114 <div class="modal fade" id="createModal" tabindex="-1">
115     <div class="modal-dialog">
116         <div class="modal-content">
117             <form action="{{ route('enrollments.store') }}" method="POST">
118                 @csrf
119                 <div class="modal-header">
120                     <h5 class="modal-title">ลงทะเบียนรายวิชาใหม่</h5>
121                     <button type="button" class="btn-close" data-bs-dismiss="modal"></button>
122                 </div>
123                 <div class="modal-body">
124                     <div class="mb-3">
125                         <label class="form-label">นักศึกษา</label>
126                         <select name="student_id" class="form-select" required>
127                             <option value="">-- เลือกนักศึกษา --</option>
128                             @foreach($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>
143                 <div class="modal-footer">
144                     <button type="button" class="btn btn-secondary" data-bs-dismiss="modal">ปิด</button>
145                     <button type="submit" class="btn btn-success">ลงทะเบียน</button>
146                 </div>
147             </form>
148         </div>
149     </div>
150 </div>
151 <!-- สิ้นสุดโมดัลเพิ่มการลงทะเบียนใหม่ -->

```

```

152
153 <!-- โมดัลเพิ่มฐานข้อมูลนักศึกษาใหม่ -->
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   <!-- สิ้นสุดโมดัลเพิ่มฐานข้อมูลนักศึกษาใหม่ -->
189 </div>
190 <!-- Bootstrap JS -->
191 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"></script>
192 </body>
193 </html>

```

\*\*\*โค้ดสามารถคัดลอกได้โดยไปที่ลิงค์นี้

<https://raw.githubusercontent.com/pyrnmsutdgt/MyB68xxxxxCRUD/refs/heads/CR/resources/views/enrollments/index.blade.php>

3.15. จากนั้น ให้ไปที่ Terminal ใน Laragon (Cmder) แล้วใช้คำสั่งต่อไปนี้  
php artisan serve

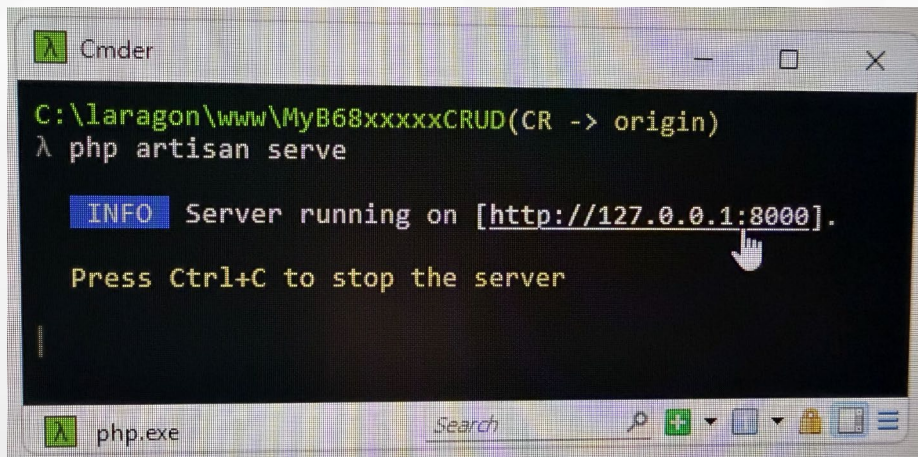


```
C:\laragon\www\MyB68xxxxxCRUD(CR -> origin)
λ php artisan serve

INFO Server running on [http://127.0.0.1:8000].

Press Ctrl+C to stop the server
```

3.16. จากนั้นให้กด ctrl + เอาลูกศรเมาส์ไปชี้ที่ [http://127.0.0.1:8000] แล้วกดที่ลิงค์นั้นเพื่อเปิดหน้าเว็บ

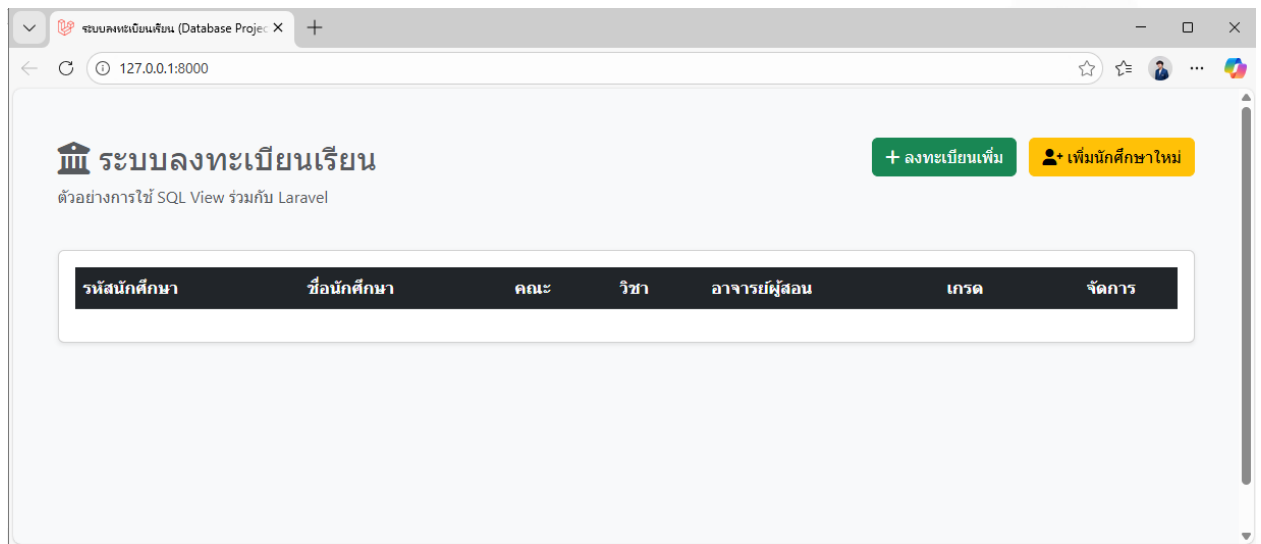


```
C:\laragon\www\MyB68xxxxxCRUD(CR -> origin)
λ php artisan serve

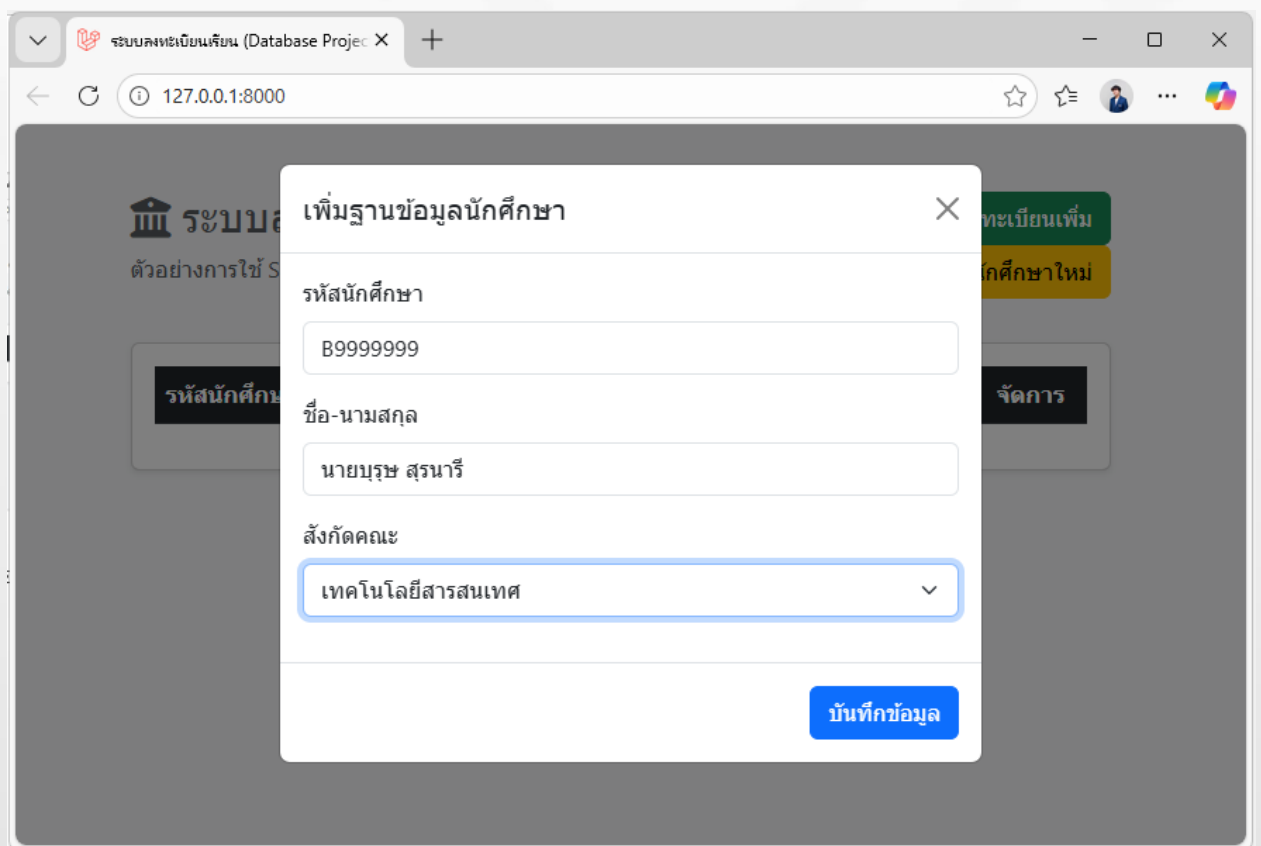
INFO Server running on [http://127.0.0.1:8000].

Press Ctrl+C to stop the server
```

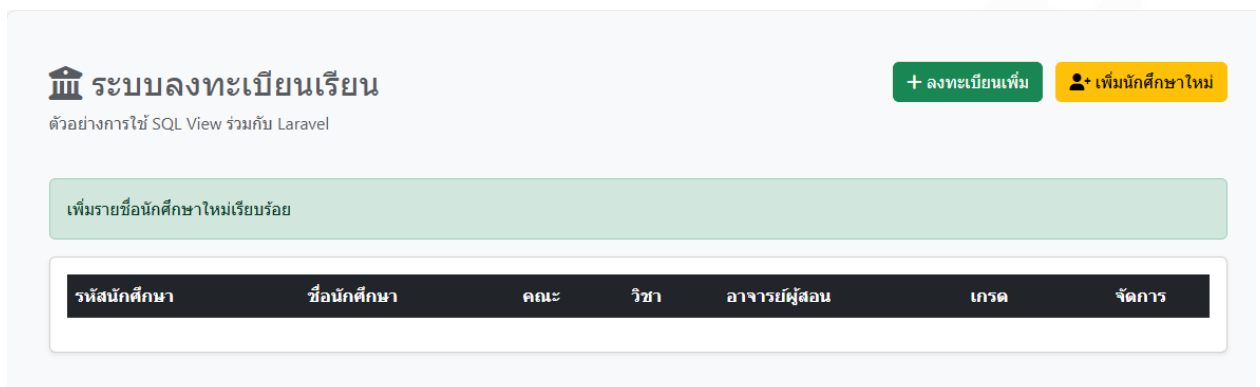
จากนั้นหน้าเว็บจะเปิดขึ้นมา (ตัวอย่างหน้าถัดไป)



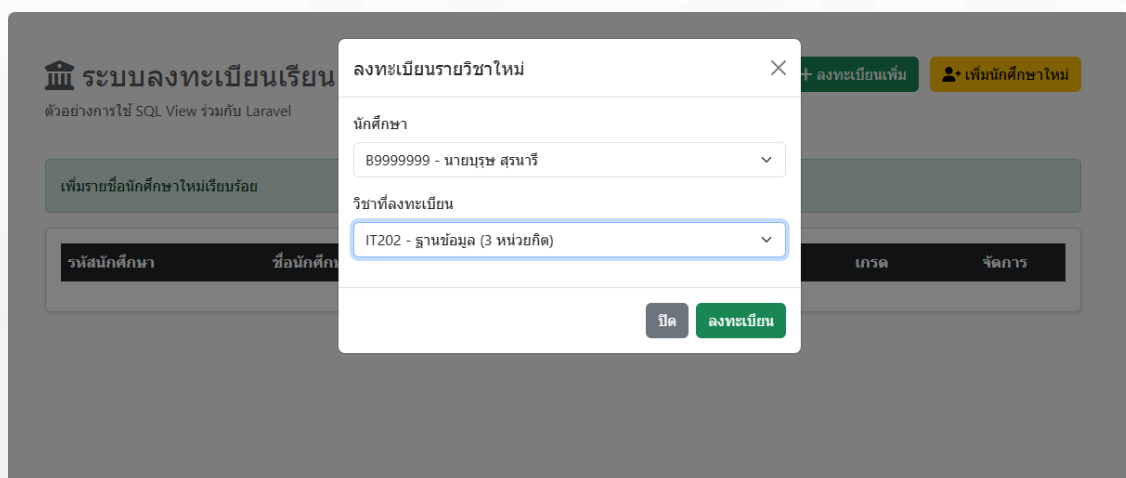
3.17. จากนั้นให้ลองเพิ่มชื่อโดยกดปุ่ม **+ เพิ่มนักศึกษาใหม่** โดยเพิ่มชื่อต่อไปนี้งไป แล้วกดบันทึกข้อมูล



หากเพิ่มได้จะมีข้อความขึ้นมา



3.18. ให้กดปุ่ม **+ ลงทะเบียนเพิ่ม** เพื่อทดลองเพิ่มและแสดงรายชื่อผู้ลงทะเบียน ดังนี้



จากนั้นกดปุ่ม ลงทะเบียน หากลงทะเบียนได้ จะขึ้นข้อความและรายชื่อขึ้นมา ดังนี้



| รหัสนักศึกษา | ชื่อนักศึกษา     | คณะ               | วิชา               | อาจารย์ผู้สอน   | เกรด   | จัดการ |
|--------------|------------------|-------------------|--------------------|-----------------|--------|--------|
| B9999999     | นายบุรุษ สุรนารี | เทคโนโลยีสารสนเทศ | IT202<br>ฐานข้อมูล | อ.วิภา รักเรียน | รอเกรด |        |



## ลายเซ็นต์ที่ 1 มีดังนี้

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

```
Cmder
C:\laragon\www\MyB68xxxxxCRUD(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\MyB68xxxxxCRUD(CR -> origin)
λ
```

**ลายเซ็นที่ 2** สามารถเพิ่มข้อมูลได้ โดยให้เพิ่มข้อมูลชื่อและรหัสนักศึกษา ตัวอย่างตามรูป

- รหัสนักศึกษา - ให้ใช้รหัสของนักศึกษา
- ชื่อ-นามสกุล - ชื่อ-นามสกุล ของนักศึกษา
- สังกัดคณะ - เลือกคณะใดก็ได้

ถ้าเพิ่มได้ เมื่อกดปุ่ม + ลงทะเบียนเพิ่ม จะมี ชื่อ-นามสกุลของนักศึกษา

**ลายเซ็นที่ 3** สามารถเพิ่มการลงทะเบียนรายวิชาได้ และแสดงผลการลงทะเบียนรายวิชาได้ ตามรายชื่อนักศึกษา ตัวอย่างตามรูป

ลงทะเบียนรายวิชาใหม่

นักศึกษา  
B6428297 - ปิยะ รองในเมือง

วิชาที่ลงทะเบียน  
IT202 - ฐานข้อมูล (3 หน่วยกิต)

ปิด ลงทะเบียน

| รหัสนักศึกษา | ชื่อนักศึกษา   |
|--------------|----------------|
| 89999999     | นายบุญ สุรนารี |

เมื่อเลือกชื่อ-นามสกุลของนักศึกษาและวิชาตามรูปแล้ว เมื่อกดปุ่มลงทะเบียน จะมีข้อความแจ้งเตือน จากนั้นข้อมูลนักศึกษาขึ้นมา ดังรูปต่อไปนี้

ลงทะเบียนเรียบร้อยแล้ว

| รหัสนักศึกษา | ชื่อนักศึกษา    | คณะ                 | วิชา            | อาจารย์ผู้สอน   | เกรด   | จัดการ |
|--------------|-----------------|---------------------|-----------------|-----------------|--------|--------|
| 89999999     | นายบุญ สุรนารี  | เทคโนโลยีสารสนเทศ   | IT202 ฐานข้อมูล | อ.วิภา รักเรียน | รอเกรด |        |
| B6428297     | ปิยะ รองในเมือง | วิทยาการคอมพิวเตอร์ | IT202 ฐานข้อมูล | อ.วิภา รักเรียน | รอเกรด |        |