

# PHIẾU HỌC TẬP CHỦ ĐỘNG

**Môn học:** CSE485: Công nghệ Web

**Họ và tên:** Nguyễn Thị Yên – MSV: 2351160567

## CHƯƠNG 8: ELOQUENT ORM

### 1. Lý thuyết Cốt lõi (Khái niệm)

- **ORM là gì?** (Object-Relational Mapping). Đây là một kỹ thuật "ánh xạ" (mapping) một bảng (table) trong CSDL (ví dụ: bảng users) thành một "Đối tượng" (Object) trong code (ví dụ: class User của PHP).
- **Eloquent (8.1):** Là ORM của Laravel.
  - **Model:** Mỗi bảng trong CSDL của bạn sẽ có một file Model tương ứng trong thư mục app/Models/. Ví dụ: Bảng sinhvien  $\rightarrow$  Model app/Models/SinhVien.php.
  - **Quy ước (Convention):** Eloquent rất thông minh. Nếu bạn đặt tên Model là SinhVien (số ít, viết hoa chữ cái đầu), nó sẽ tự động hiểu rằng Model này làm việc với bảng sinhvien (số nhiều, viết thường).
- **Migrations (8.2):** Là "hệ thống quản lý phiên bản" cho CSDL của bạn. Thay vì tạo bảng bằng tay trong phpMyAdmin, bạn sẽ viết code PHP (trong thư mục database/migrations/) để định nghĩa cấu trúc bảng.
  - Lệnh tạo: `php artisan make:model TenModel -m` (cờ -m sẽ tạo luôn file migration).
  - Lệnh chạy: `php artisan migrate` (để tạo bảng trong CSDL).
- **Truy vấn (8.3, 8.4):** Đây là điều tuyệt vời nhất.

- Thay vì PDO (Chương 4):

```
$sql = "SELECT * FROM sinhvien";
```

- Dùng Eloquent (Chương 8):

```
$all_students = SinhVien::all();
```

- Thay vì INSERT (Chương 4):

```
$sql = "INSERT INTO sinhvien (ten, email) VALUES (?, ?)";
```

- Dùng Eloquent (Chương 8):

```
SinhVien::create(['ten_sinh_vien' => $ten, 'email' => $email]);
```

## 2.Nhiệm vụ Thực hành (BẮT BUỘC)

**Kịch bản:** Chúng ta sẽ "nâng cấp" PHT Quản lý sinh viên (từ Chương 4 & 5) lên chuẩn Laravel 100%, sử dụng Eloquent Model và Migrations.

---

### A. Thiết lập Database & Migration (8.2)

1. // **TODO 1:** Mở file .env ở thư mục gốc dự án (cse485\_chapter6).

2. // **TODO 2:** Sửa các thông số CSDL. Tạo một CSDL mới trong phpMyAdmin tên là cse485\_laravel (hoặc tên khác tùy bạn).

Đoạn mã

```
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=cse485_laravel # Tên CSDL bạn vừa tạo
DB_USERNAME=root          # User XAMPP
DB_PASSWORD=              # Pass XAMPP (rỗng)
```

3. // **TODO 3:** Mở Terminal, cd vào dự án, chạy lệnh Artisan sau để tạo Model SinhVien VÀ file migration (-m): Bash php artisan make:model SinhVien -m

4. // **TODO 4:** Mở file migration vừa được tạo trong: database/migrations/...\_create\_sinh\_viens\_table.php (tên file có thêm timestamp).

5. // **TODO 5:** Sửa hàm up() để định nghĩa cấu trúc bảng (thêm 2 cột ten\_sinh\_vien và email): PHP

```
public function up(): void
{
```

```
Schema::create('sinh_viens', function (Blueprint $table) {  
    $table->id(); // Tự động tạo cột 'id' (bigint, auto-increment, primary key)  
  
    // TODO 5: Thêm 2 dòng này  
    $table->string('ten_sinh_vien', 255);  
    $table->string('email', 255)->unique(); // unique() là ràng buộc duy nhất  
  
    $table->timestamps(); // Tự động tạo 2 cột 'created_at' và 'updated_at'  
});  
}
```

**6. // TODO 6:** Chạy lệnh migrate để tạo bảng trong

CSDL: Bash php artisan migrate

(Vào phpMyAdmin kiểm tra CSDL cse485\_laravel, bạn sẽ thấy bảng sinh\_viens xuất hiện).

---

## B. Cập nhật Model & Controller (8.3, 8.4)

**1. // TODO 7:** Mở file Model app/Models/SinhVien.php.

**2. // TODO 8:** Để cho phép ::create() (Mass Assignment), chúng ta phải khai báo use HasFactory; và các trường được phép điền vào mảng \$fillable:

PHP

```

<?php namespace App\Models; use
Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class SinhVien extends
Model
{
    use
    HasFactory;

    // TODO 8: Thêm mảng $fillable
    protected $fillable = [
        'ten_sinh_vien',
        'email',
    ];
}

```

**3. // TODO 9:** (Tùy chọn) Tạo 1 Controller mới cho Sinh Viên: Bash php

```
artisan make:controller SinhVienController
```

**4. // TODO 10:** Mở app/Http/Controllers/SinhVienController.php và use Model SinhVien ở đầu file. Sau đó tạo 2 phương thức index() (để hiển thị) và store() (để lưu):

PHP

```
<?php namespace  
App\Http\Controllers; use  
Illuminate\Http\Request;  
  
// TODO 10: Import Model SinhVien  
use App\Models\SinhVien;
```

```
class SinhVienController extends Controller
{
    // Phương thức index() (SELECT)
    public function index()
    {
        // TODO 11: Dùng Eloquent ::all() để lấy toàn bộ sinh viên
        // Gợi ý: $danhsachSV = SinhVien::all();

        // TODO 12: Trả về 1 view 'sinhvien.list' và truyền $danhsachSV
        // Gợi ý: return view('...', compact('...'));
    }
    // Phương thức store() (INSERT)    public function store(Request $request)
    {
        // TODO 13: Lấy toàn bộ dữ liệu từ form
        // Gợi ý: $data = $request->all();

        // TODO 14: Dùng Eloquent ::create() để lưu vào CSDL
        // (Lưu ý: tên input trong form phải khớp với $fillable và tên cột)
        // Gợi ý: SinhVien::create($data);

        // TODO 15: Chuyển hướng về trang danh sách
        // Gợi ý: return redirect()->route('sinhvien.index');
    }
}
```

**5. // TODO 16:** Mở routes/web.php, import SinhVienController và tạo 2 route (1 GET, 1 POST): PHP

```
// Gợi ý:  
  
// use App\Http\Controllers\SinhVienController;  
  
// Route::get('/sinhvien', [SinhVienController::class, 'index'])->name('sinhvien.index');  
  
// Route::post('/sinhvien', [SinhVienController::class, 'store'])->name('sinhvien.store');
```

**6. // TODO 17:** Tạo file View resources/views/sinhvien/list.blade.php (tạo thư mục sinhvien nếu chưa có). Copy/paste code từ PHT Chương 7 (layout, form, table) và sửa lại:

- o Form: action="{{ route('sinhvien.store') }}" và method="POST". Thêm @csrf (Bắt buộc cho form POST của Laravel - sẽ học ở Chương 9). Input name phải là ten\_sinh\_vien và email.
- o Table: Dùng @foreach(\$danhsachSV as \$sv) và  
{{ \$sv->ten\_sinh\_vien }}.

### 3. Yêu cầu Bằng chứng (Proof of Work)

Bạn phải nộp lại 3 bằng chứng sau:

#### A. Code đã hoàn thiện:

1. Dán (paste) code hàm up() trong file Migration (...\_create\_sinh\_viens\_table.php).

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

```

return new class extends Migration
{
    /**
     * Run the migrations.
     */
    public function up(): void
    {
        Schema::create('sinh_viens', function (Blueprint $table) {
            $table->id(); // Tự động tạo cột 'id' (bigint, auto-increment, primary
key)

            // TODO 5: Thêm 2 dòng này
            $table->string('ten_sinh_vien', 255);
            $table->string('email', 255)->unique(); // unique() là ràng buộc duy
nhất

            $table->timestamps(); // // Tự động tạo 2 cột 'created_at' và
'updated_at'

        });
    }

    /**
     * Reverse the migrations.
     */
    public function down(): void
    {

```

```
Schema::dropIfExists('sinh_viens');  
  
}  
  
};
```

2. Dán (paste) toàn bộ code file **app/Models/SinhVien.php**.

```
<?php  
namespace App\Models;  
use Illuminate\Database\Eloquent\Factories\HasFactory;  
use Illuminate\Database\Eloquent\Model;  
class SinhVien extends Model  
{  
    use HasFactory;  
    // TODO 8: Thêm mảng $fillable  
    protected $fillable = [  
        'ten_sinh_vien',  
        'email',  
    ];  
}
```

3. Dán (paste) toàn bộ code file  
**app/Http/Controllers/SinhVienController.php**.

```
<?php  
namespace App\Http\Controllers;  
use Illuminate\Http\Request;  
// TODO 10: Import Model SinhVien
```

```
use App\Models\SinhVien;

class SinhVienController extends Controller
{
    // Phương thức index() (SELECT)
    public function index()
    {
        $title = "PHT Chương 8 - Quản lý Sinh Viên";
        // TODO 11: Dùng Eloquent ::all() để lấy toàn bộ sinh viên
        $danhsachSV = SinhVien::all();
        // TODO 12: Trả về 1 view 'sinhvien.list' và truyền $danhsachSV
        return view('sinhvien.list', compact('danhsachSV', 'title'));
    }

    // Phương thức store() (INSERT)
    public function store(Request $request)
    {
        // TODO 13: Lấy toàn bộ dữ liệu từ form
        $data = $request->all();
        // TODO 14: Dùng Eloquent ::create() để lưu vào CSDL
        // (Lưu ý: tên input trong form phải khớp với $fillable và tên cột)
        SinhVien::create($data);
        // TODO 15: Chuyển hướng về trang danh sách
        return redirect()->route('sinhvien.index');
    }
}
```

4. Dán (paste) code 2 route trong **routes/web.php**.

```
<?php

use Illuminate\Support\Facades\Route;

use App\Http\Controllers\PageController;

use App\Http\Controllers\SinhVienController;

// chương 6

Route::get('/', [PageController::class, 'showHomepage']);

Route::get('/about', [PageController::class, 'showHomepage']);


// chương 8

Route::get('/sinhvien', [SinhVienController::class,
'index'])->name('sinhvien.index');

Route::post('/sinhvien', [SinhVienController::class,
'store'])->name('sinhvien.store');
```

## **B. Ảnh chụp màn hình Kết quả (3 ẢNH):**

- 1. Ảnh 1 (Terminal):** Chụp màn hình Terminal sau khi chạy php artisan make:model ... -m VÀ php artisan migrate thành công.

```
Admin@nguyenvn MINGW64 /d/xampp/htdocs/cse485_chapter6
$ php artisan migrate
[INFO] Nothing to migrate.

Admin@nguyenvn MINGW64 /d/xampp/htdocs/cse485_chapter6
$ php artisan make:model SinhVien -m
[INFO] Model [D:\xampp\htdocs\cse485_chapter6\app\Models\SinhVien.php] created successfully.
[INFO] Migration [D:\xampp\htdocs\cse485_chapter6\database\migrations\2025_12_16_160642_create_sinh_viens_table.php] created successfully.

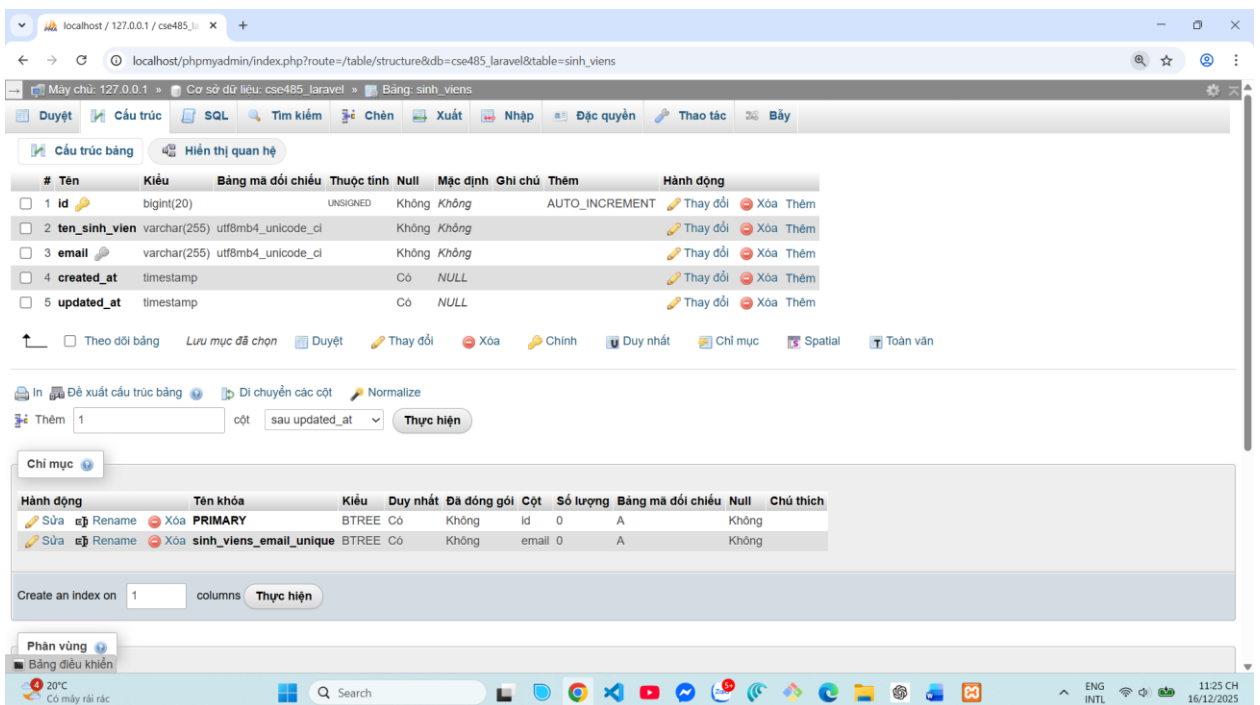
Admin@nguyenvn MINGW64 /d/xampp/htdocs/cse485_chapter6
$ php artisan migrate
[WARN] The database 'cse485_laravel' does not exist on the 'mysql' connection.

Would you like to create it? (yes/no) [yes]
yes
[INFO] Preparing database.
Creating migration table ..... 20.57ms DONE
[INFO] Running migrations.
0001_01_01_000000_create_users_table ..... 78.16ms DONE
0001_01_01_000001_create_cache_table ..... 18.87ms DONE
0001_01_01_000002_create_jobs_table ..... 67.75ms DONE
2025_12_16_160642_create_sinh_viens_table ..... 29.71ms DONE

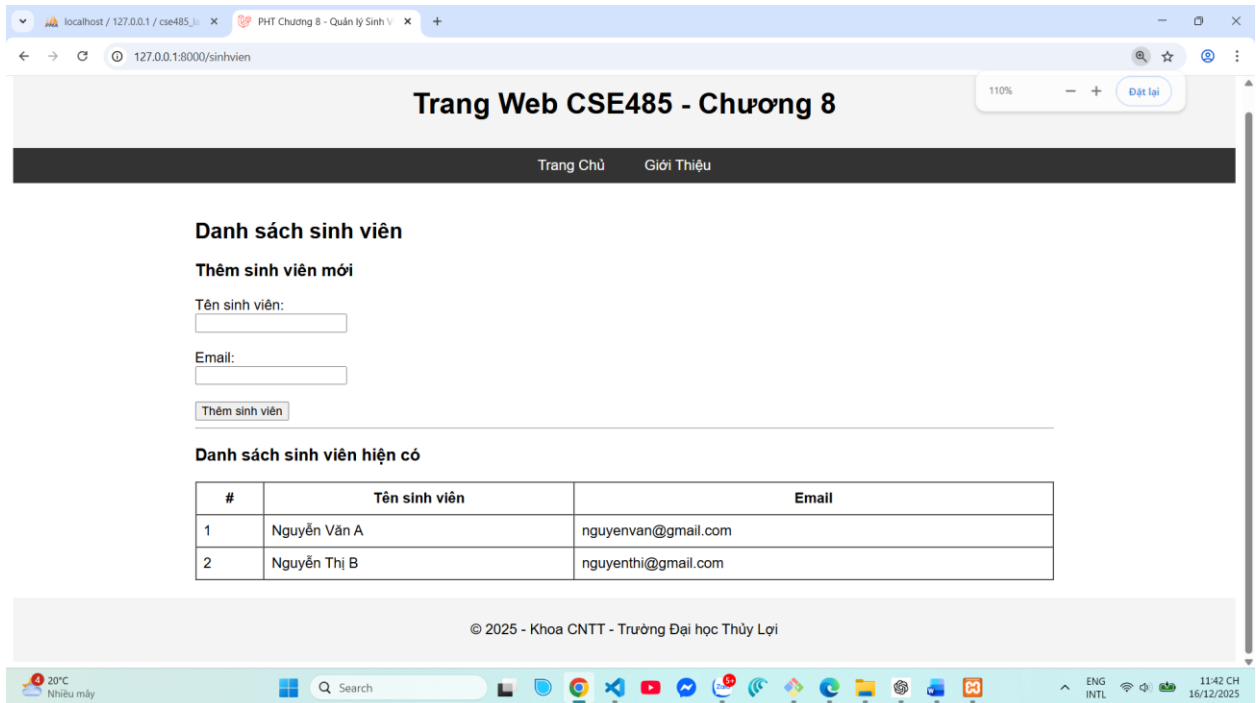
Admin@nguyenvn MINGW64 /d/xampp/htdocs/cse485_chapter6
$ php artisan make:controller SinhVienController
[INFO] Controller [D:\xampp\htdocs\cse485_chapter6\app\Http\Controllers\SinhVienController.php] created successfully.

Admin@nguyenvn MINGW64 /d/xampp/htdocs/cse485_chapter6
$ |
```

2. Ảnh 2 (phpMyAdmin): Chụp màn hình tab "Structure" (Cấu trúc) của bảng sinh\_viens trong CSDL, cho thấy rõ các cột id, ten\_sinh\_vien, email, created\_at.



3. Ảnh 3 (Trình duyệt Web): Chụp ảnh màn hình trang /sinhvien, sau khi đã dùng form thêm 1-2 sinh viên (chứng minh ::create() và ::all() đều hoạt động).



#### 4. Câu hỏi Phản biện (Bắt buộc)

Sau khi hoàn thành Phần 2 & 3, hãy đặt 01 câu hỏi tư duy.

(Gợi ý: "Hãy so sánh PHT này (Eloquent) với PHT Chương 4 (PDO). Lợi ích lớn nhất của Eloquent là gì? Khi nào (trường hợp nào) thì dùng Eloquent sẽ tiện lợi hơn, và khi nào thì dùng PDO/SQL thuần có thể sẽ tốt hơn?").

**Câu hỏi của tôi là:**

1. Điều gì xảy ra nếu sửa Migration **sau khi đã migrate**?
2. Eloquent có chậm hơn SQL thuần không? Vì sao?
3. Nếu cần truy vấn phức tạp (JOIN nhiều bảng), Eloquent có còn phù hợp?
4. Nếu đổi tên cột trong bảng, bạn cần sửa những gì trong Laravel?

#### 5. Kết nối Đánh giá (Rất quan trọng)

**Eloquent ORM** (Chương 8) là "trái tim" xử lý dữ liệu của Laravel. Kỹ năng `make:model`, `migrate`, và sử dụng các phương thức `::all()`, `::find()`, `::create()` là **tối quan trọng**.

Đây là kiến thức **trọng tâm** của cả Bài tập trên lớp (Phần Laravel) (20%)<sup>2</sup> và **Bài tập lớn theo nhóm (50%)**<sup>3</sup>. Nếu không nắm vững Eloquent, bạn không thể xây dựng bất kỳ chức năng nào cho dự án của mình.