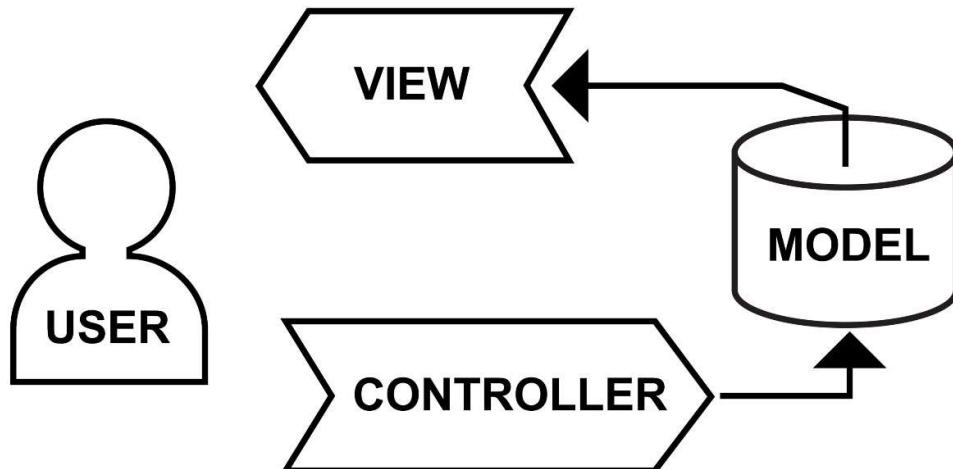


## CHƯƠNG 5: MỞ RỘNG ỨNG DỤNG (TẬP TRUNG VÀO 5.1 TÁI CẤU TRÚC & 5.5 MÔ HÌNH MVC)

### 1. Lý thuyết Cốt lõi (Khái niệm)

Khái niệm cốt lõi của chương này là **Mô hình MVC (Model - View - Controller)**. Đây là một mẫu kiến trúc (architecture pattern) giúp tách biệt ứng dụng của bạn thành 3 thành phần riêng biệt, mỗi thành phần chỉ làm một việc duy nhất:

- **Model (M):** Tầng "Dữ liệu".
  - Chịu trách nhiệm **duy nhất** cho mọi tương tác với CSDL (truy vấn, thêm, sửa, xóa).
  - Nó không biết gì về HTML hay cách hiển thị. Nó chỉ nhận lệnh (ví dụ: "lấy 10 sinh viên") và trả về dữ liệu thô.
  - Trong PHT của chúng ta, đây là nơi chứa code PDO, SELECT, INSERT.
- **View (V):** Tầng "Hiển thị".
  - Chịu trách nhiệm **duy nhất** cho việc hiển thị giao diện người dùng (HTML, CSS).
  - Nó **không** được phép truy vấn CSDL. Nó chỉ nhận dữ liệu (do Controller đưa) và echo ra màn hình.
  - Đây là nơi chứa code <table>, <form>, foreach để lặp và echo.
- **Controller (C):** Tầng "Điều khiển" (Bộ não).
  - Là "nhạc trưởng" điều phối mọi thứ.
  - Nó nhận yêu cầu từ người dùng (ví dụ: \$\_POST từ form).
  - Nó "ra lệnh" cho Model (ví dụ: "Thêm sinh viên này vào CSDL" hoặc "Lấy tất cả sinh viên").
  - Nó nhận kết quả từ Model, và "truyền" dữ liệu đó cho View để hiển thị.



**MODEL - VIEW - CONTROLLER PATTERN**

## 2. Nhiệm vụ Thực hành (BẮT BUỘC)

**Kịch bản:** Chúng ta sẽ **Tái cấu trúc (Refactor)** PHT Chương 4 (Quản lý sinh viên) thành mô hình MVC.

**Cấu trúc thư mục (Bắt buộc):** Tạo một thư mục chapter5/ và cấu trúc 3 tệp + 2 thư mục con như sau:

```
/chapter5/
| -- index.php          (Đây là Controller)
| -- models/
|   | -- SinhVienModel.php (Đây là Model)
| -- views/
|   | -- sinhvien_view.php (Đây là View)
```

---

### Code Khởi đầu (Starter Code):

#### Tập 1: models/SinhVienModel.php (Model)

PHP

```
<?php

// Tệp Model sẽ chứa tất cả logic truy vấn CSDL


// TODO 1: Viết 1 hàm tên là getAllSinhVien()
// Hàm này nhận 1 tham số là $pdo (đối tượng PDO)
// Bên trong hàm, thực thi câu lệnh SELECT * FROM sinhvien
// Hàm trả về kết quả (dùng fetchAll)

function getAllSinhVien($pdo) {
    // Gợi ý:
    // $sql = "SELECT * FROM sinhvien";
    // $stmt = $pdo->query($sql);
    // return $stmt->fetchAll(PDO::FETCH_ASSOC);
}

// TODO 2: Viết 1 hàm tên là addSinhVien()
// Hàm này nhận 3 tham số: $pdo, $ten, $email
// Bên trong hàm, thực thi câu lệnh INSERT (dùng Prepared Statement)
```

```
// Gợi ý:  
// $sql = "INSERT INTO sinhvien (ten_sinh_vien, email) VALUES (?, ?)";  
// $stmt = $pdo->prepare($sql);  
// $stmt->execute([$ten, $email]);  
  
?>
```

## Tệp 2: views/sinhvien\_view.php (View)

PHP

```
<?php  
// Tệp View CHỈ chứa HTML và logic hiển thị (echo, foreach)  
// Tệp View KHÔNG chứa câu lệnh SQL  
?>  
<!DOCTYPE html>  
<html lang="vi">  
<head>  
    <meta charset="UTF-8">  
    <title>PHT Chương 5 - MVC</title>  
    <style>  
        table { width: 100%; border-collapse: collapse; }  
        th, td { border: 1px solid #ddd; padding: 8px; }  
        th { background-color: #f2f2f2; }  
    </style>  
</head>  
<body>  
    <h2>Thêm Sinh Viên Mới (Kiến trúc MVC)</h2>  
  
    <h2>Danh Sách Sinh Viên (Kiến trúc MVC)</h2>  
    <table>  
        <tr>  
            <th>ID</th>  
            <th>Tên Sinh Viên</th>  
            <th>Email</th>  
            <th>Ngày Tạo</th>  
        </tr>
```

```

<?php

// TODO 4: Dùng vòng lặp foreach để duyệt qua biến $danh_sach_sv
// (Biến $danh_sach_sv này sẽ được Controller truyền sang)
// Gợi ý: foreach ($danh_sach_sv as $sv) { ... }

// TODO 5: In (echo) các dòng <tr> và <td> chứa dữ liệu $sv
// Gợi ý: echo "<tr><td>" . htmlspecialchars($sv['id']) .
"</td>...</tr>";

// Đóng vòng lặp

?>
</table>
</body>
</html>

```

### Tệp 3: index.php (Controller)

PHP

```

<?php

// Tệp Controller là "não" của ứng dụng

// TODO 6: (Quan trọng) Import (require_once) tệp Model vào
// Gợi ý: require_once 'models/SinhVienModel.php';

// === THIẾT LẬP KẾT NỐI PDO ===

// TODO 7: Copy code PDO từ PHT Chương 4 vào đây
$host = '127.0.0.1';
$dbname = 'cse485_web';
$username = 'root';
$password = '';
$dsn = "mysql:host=$host;dbname=$dbname;charset=utf8mb4";
try {
    $pdo = new PDO($dsn, $username, $password);
    $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
}

```

```
    } catch (PDOException $e) {
        die("Kết nối thất bại: " . $e->getMessage());
    }
    // === KẾT THÚC KẾT NỐI PDO ===

    // === LOGIC CỦA CONTROLLER ===

    // TODO 8: Kiểm tra xem có hành động POST (thêm sinh viên) không
    // Gợi ý: Dùng isset($_POST['ten_sinh_vien'])

    if ( ... ) {

        // TODO 9: Nếu có, lấy $ten và $email từ $_POST
        $ten = ...;
        $email = ...;

        // TODO 10: Gọi hàm addSinhVien() từ Model
        // (Truyền $pdo, $ten, $email vào hàm)
        // Gợi ý: addSinhVien($pdo, $ten, $email);

        // TODO 11: Chuyển hướng về index.php để "làm mới" trang
        // Gợi ý: header('Location: index.php');
        exit;
    }

    // TODO 12: (Luôn luôn) Gọi hàm getAllSinhVien() từ Model
    // (Truyền $pdo vào hàm)
    // Lưu kết quả trả về vào một biến, ví dụ: $danh_sach_sv
    // Gợi ý: $danh_sach_sv = getAllSinhVien($pdo);

    // TODO 13: (Rất quan trọng) Import (include) tệp View ở cuối cùng
    // Tệp View sẽ tự động "nhìn thấy" biến $danh_sach_sv mà ta vừa tạo
    // Gợi ý: include 'views/sinhvien_view.php';
```

?>

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

Bạn phải nộp lại 2 bằng chứng sau:

**A. Code đã hoàn thiện:** Dán (paste) toàn bộ code của cả 3 tệp bạn đã sửa:

1. models/SinhVienModel.php
2. views/sinhvien\_view.php
3. index.php

**B. Ảnh chụp màn hình Kết quả (Trình duyệt Web):** Chạy tệp index.php của bạn trên XAMPP (ví dụ: <http://localhost/chapter5/>) và chụp ảnh màn hình trình duyệt. *Lưu ý: Kết quả trông phải giống hệt PHT Chương 4 (chứng minh việc tái cấu trúc đã thành công mà không làm thay đổi giao diện).*

(Dán Code A1, A2, A3 và Ảnh B của bạn vào đây)

### 4. Câu hỏi Phản biện (Bắt buộc)

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

(Gợi ý: "Trong file views/sinhvien\_view.php, tại sao chúng ta **tuyệt đối không** được phép viết code kết nối PDO hay SELECT \*? Lợi ích của việc 'cầm' View làm việc trực tiếp với CSDL là gì?").

**Câu hỏi của tôi là:** (Bạn tự điền câu hỏi của mình vào đây)

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

Mô hình **MVC** (Chủ đề 5.5) là kiến trúc **BẮT BUỘC** và là nền tảng của toàn bộ **Laravel Framework** mà chúng ta sẽ học từ Chương 6.

Hiểu rõ cách tách biệt Model - View - Controller trong PHT này sẽ quyết định bạn có học được Laravel hay không. Kỹ năng này sẽ được áp dụng trực tiếp trong:

- **Bài tập trên lớp (Phần Laravel - 20%)**
- **Bài tập lớn theo nhóm (50%)**