

Phiếu học tập chủ động
Môn học: CSE485 – Công Nghệ Web

Họ và tên: Nguyễn Đức Quân

Mã sinh viên: 2251162122

Lớp: 64HTTT4

Lớp học phần: 65HTTT

CHƯƠNG 5: MỞ RỘNG ỨNG DỤNG

CODE

MODELS;

```
<?php
```

```
// =====
```

```
// MODEL: Chứa logic truy vấn CSDL cho bảng sinhvien
```

```
// =====
```

```
// === KẾT NỐI CSDL ===
```

```
$host = '127.0.0.1';
```

```
$dbname = 'pht_05';
```

```
$username = 'root';
```

```
$password = ";
```

```
$dsn = "mysql:host=$host;dbname=$dbname;charset=utf8mb4";
```

```
try {
```

```
    $pdo = new PDO($dsn, $username, $password);
```



```
// Thiết lập chế độ thông báo lỗi
$pdo->setAttribute(PDO::ATTR_ERRMODE,
PDO::ERRMODE_EXCEPTION);

} catch (PDOException $e) {
    die("Kết nối thất bại: " . $e->getMessage());
}

// =====
// TODO 1: Hàm lấy toàn bộ sinh viên
// =====

function getAllSinhVien(PDO $pdo) {
    try {
        $sql = "SELECT * FROM sinhvien";
        $stmt = $pdo->query($sql);

        return $stmt->fetchAll(PDO::FETCH_ASSOC);
    } catch (PDOException $e) {
        return [];
    }
}

// =====
// TODO 2: Hàm thêm sinh viên
```

```

// Tham số: $ten, $email
// =====

function addSinhVien(PDO $pdo, string $ten, string $email) {
    try {
        $sql = "INSERT INTO sinhvien (ten_sinh_vien, email) VALUES (?, ?)";
        $stmt = $pdo->prepare($sql);

        // Truyền dữ liệu vào câu SQL
        return $stmt->execute([$ten, $email]); // true/false
    } catch (PDOException $e) {
        return false;
    }
?>
```

VIEWS

```

<?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; }  
    tr:nth-child(even) { background-color: #fafafa; }  
</style>
```

```
</head>
```

```
<body>
```

```
    <h2>Thêm Sinh Viên Mới (Kiến trúc MVC)</h2>
```

```
    <!-- Form thêm sinh viên (Controller sẽ xử lý POST) -->
```

```
    <form method="post" action="">  
        <label>  
            Tên sinh viên:  
            <input type="text" name="ten_sinh_vien" required>  
        </label>  
        <br><br>  
        <label>  
            Email:  
            <input type="email" name="email" required>  
        </label>  
        <br><br>
```

```
<button type="submit" name="action" value="add">Thêm sinh viên</button>
</form>
```

```
<hr>
```

```
<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)
```

```
if (!empty($danh_sach_sv) && is_array($danh_sach_sv)) {
```

```
    foreach ($danh_sach_sv as $sv) {
```

```
        echo '<tr>';
```

```
        echo '<td>' . htmlspecialchars($sv['id'] ?? '') . '</td>';
```

```
        echo '<td>' . htmlspecialchars($sv['ten_sinh_vien'] ?? '') . '</td>';
```

```
        echo '<td>' . htmlspecialchars($sv['email'] ?? '') . '</td>';
```

```

// Tùy tên cột ngày trong DB: created_at / ngay_tao ...
echo '<td>' . htmlspecialchars($sv['created_at'] ?? ($sv['ngay_tao'] ?? '')) .
. '</td>';
echo '</tr>';
}

} else {
    echo '<tr><td colspan="4">Không có sinh viên nào trong hệ
thông.</td></tr>';
}

?>

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

```

INDEX.PHP

```

<?php
// =====
// Controller chính cho Sinh Viên
// =====

// Nạp Model (bên trong có cả kết nối PDO và các hàm thao tác CSDL)
require_once 'models/SinhvienModel.php';

// Biến dùng để lưu thông báo (nếu cần hiển thị ra View)
$message = "";

```

```
// =====  
  
// Xử lý submit Form thêm sinh viên  
// =====  
  
if ($_SERVER['REQUEST_METHOD'] === 'POST') {  
    // Kiểm tra có nút action = add được gửi lên hay không  
    if (isset($_POST['action']) && $_POST['action'] === 'add') {  
  
        // Lấy dữ liệu từ form  
        $ten = isset($_POST['ten_sinh_vien']) ? trim($_POST['ten_sinh_vien']) : " ";  
        $email = isset($_POST['email']) ? trim($_POST['email']) : " ";  
  
        // Validate đơn giản  
        if ($ten === "" || $email === "") {  
            $message = "Vui lòng nhập đầy đủ Tên và Email.";  
        } else {  
            // Gọi hàm Model để thêm sinh viên  
            $ok = addSinhVien($pdo, $ten, $email);  
  
            if ($ok) {  
                $message = "Thêm sinh viên thành công!";  
                // Có thể redirect để tránh lỗi F5 gửi lại form:  
                // header("Location: index.php");  
                // exit;  
            }  
        }  
    }  
}
```

```
    } else {
        $message = "Có lỗi xảy ra, không thể thêm sinh viên.";
    }
}

}

// =====
// Luôn luôn lấy danh sách sinh viên
// =====
$danh_sach_sv = getAllSinhVien($pdo);

// =====
// Gọi View để hiển thị
// =====

// Biến $danh_sach_sv (và $message nếu bạn muốn dùng) sẽ được sử dụng trong
View
require 'views/sinhvien_view.php';
```

ẢNH CHỤP MÀN HÌNH

The screenshot shows a web browser window with the URL `localhost/CSE485/Chapter5/`. The page has two main sections:

- Thêm Sinh Viên Mới (Kiến trúc MVC)**: A form with fields for 'Tên sinh viên:' and 'Email:', and a button 'Thêm sinh viên'.
- Danh Sách Sinh Viên (Kiến trúc MVC)**: A table listing three students with columns: ID, Tên Sinh Viên, Email, and Ngày Tạo.

ID	Tên Sinh Viên	Email	Ngày Tạo
1	Nguyễn Đức Quán	nguyennquuan06@gmail.com	2025-12-01 11:10:04
2	Nguyễn Anh Tuấn	aTuan@gmail.com	2025-12-01 11:10:12
3	Nguyễn Anh Tuấn	aTuan@gmail.com	2025-12-01 20:49:42

CÂU HỎI PHẢN BIỆN

Câu hỏi: Trong mô hình MVC, tại sao Controller không nên viết trực tiếp HTML, và View không nên xử lý dữ liệu (vd: tính toán, truy vấn DB)?

Trả lời:

- **1. Controller không được viết HTML**
 - Controller chỉ đóng vai trò “điều phối”, nhận yêu cầu, gọi Model, gửi dữ liệu sang View.
 - Nếu Controller viết HTML → code lẩn lộn, khó bảo trì, khó mở rộng.
 - Mọi thay đổi giao diện sẽ phải sửa trong Controller → sai kiến trúc MVC.
- **2. View không được xử lý dữ liệu hoặc truy vấn CSDL**
 - View chỉ để hiển thị thông tin ra giao diện người dùng.
 - Nếu View xử lý logic, tính toán hoặc truy vấn → giao diện trở nên nặng, khó test, khó tái sử dụng.
 - Việc tách riêng giúp View chỉ cần nhận dữ liệu đã “xử lý sạch”.
- **Lợi ích chung**
 - Code rõ ràng, dễ bảo trì
 - Tăng tính tái sử dụng
 - Giảm lỗi do trộn lẫn logic và giao diện
 - Cùng một Model/Controller có thể dùng cho nhiều View khác nhau (web, mobile, API)

