

CHƯƠNG 4: WEBSITE HƯỚNG DẪN LIỆU

1. Lý thuyết Cốt lõi (Khái niệm)

Để PHP "nói chuyện" với MySQL (hoặc bất kỳ CSDL nào), chúng ta sử dụng một giao diện (interface) chuẩn gọi là PDO (PHP Data Objects).

Luồng làm việc với PDO luôn gồm các bước:

1. Kết nối (Connect): Tạo một đối tượng PDO mới, cung cấp cho nó "chuỗi kết nối" (DSN), username và password của CSDL.
2. Chuẩn bị (Prepare): Viết câu lệnh SQL (như SELECT * FROM users WHERE id = ?). Dấu ? là một placeholder (trình giữ chỗ).
3. Thực thi (Execute): "Bind" (gắn) giá trị thật (ví dụ: \$id = 5) vào placeholder ? và thực thi câu lệnh.
4. Lấy kết quả (Fetch): Nếu là câu SELECT, dùng fetch() (lấy 1 dòng) hoặc fetchAll() (lấy tất cả) để nhận dữ liệu.

Tại sao dùng "Prepared Statements" (dấu "?")? Đây là cách bắt buộc để chống lại một kiểu tấn công cực kỳ phổ biến tên là SQL Injection. Tuyệt đối không bao giờ viết code bằng cách cộng chuỗi trực tiếp như: \$sql = "SELECT * FROM users WHERE username = " . \$_POST['user'] . """; (CỰC KỲ NGUY HIỂM!)

2. Nhiệm vụ Thực hành (BẮT BUỘC)

Kịch bản: Xây dựng một trang "Danh sách sinh viên" đơn giản. Trang này cho phép bạn:

1. Thêm sinh viên mới vào CSDL (Dùng INSERT).
2. Hiển thị toàn bộ sinh viên đang có trong CSDL (Dùng SELECT).

A. Thiết lập Ban đầu (Bắt buộc)

1. Mở phpMyAdmin.
2. Tạo một CSDL mới tên là cse485_web.
3. Chọn CSDL cse485_web, mở tab SQL và chạy lệnh sau để tạo bảng: SQL

```
CREATE TABLE sinhvien (
    id INT AUTO_INCREMENT PRIMARY KEY,
    ten_sinh_vien VARCHAR(255) NOT NULL,
    email VARCHAR(255) NOT NULL,
    ngay_tao TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

B. Code Khởi đầu (Starter Code):

Tạo 1 tệp chapter4.php trong thư mục htdocs của XAMPP:

PHP

```
<?php

// === THIẾT LẬP KẾT NỐI PDO ===

$host = "localhost"; // hoặc localhost

$dbname = 'cse485_web'; // Tên CSDL bạn vừa tạo

$username = 'root'; // Username mặc định của XAMPP

$password = ""; // Password mặc định của XAMPP (rỗng)

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

try {

    // TODO 1: Tạo đối tượng PDO để kết nối CSDL

    // Gợi ý: $pdo = new PDO(...);

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

    $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

    // echo "Kết nối thành công!"; // (Bỏ comment để test)

} catch (PDOException $e) {

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

}

// === LOGIC THÊM SINH VIÊN (XỬ LÝ FORM POST) ===

// TODO 2: Kiểm tra xem form đã được gửi đi (method POST) và có 'ten_sinh_vien' không

if (isset($_POST['ten_sinh_vien']) && isset($_POST['email'])) {

    // TODO 3: Lấy dữ liệu 'ten_sinh_vien' và 'email' từ $_POST

    $ten = $_POST['ten_sinh_vien'];

    $email = $_POST['email'];

    // TODO 4: Viết câu lệnh SQL INSERT với Prepared Statement (dùng dấu ?)

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

    // TODO 5: Chuẩn bị (prepare) và thực thi (execute) câu lệnh
```

```

$stmt = $pdo->prepare($sql);

$stmt->execute([$ten, $email]);

// TODO 6: (Tùy chọn) Chuyển hướng về chính trang này để "làm mới"
header('Location: chapter4.php');

exit;

}

// === LOGIC LẤY DANH SÁCH SINH VIÊN (SELECT) ===

$sql_select = "SELECT * FROM sinhvien ORDER BY ngay_tao DESC";
$stmt_select = $pdo->query($sql_select);

?>

<!DOCTYPE html>

<html lang="vi">

<head>

<meta charset="UTF-8">

<title>PHT Chương 4 - Website hướng dẫn liệu</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 (Chủ đề 4.3)</h2>

<form action="chapter4.php" method="POST">

    Tên sinh viên: <input type="text" name="ten_sinh_vien" required>

    Email: <input type="email" name="email" required>

    <button type="submit">Thêm</button>

</form>

<h2>Danh Sách Sinh Viên (Chủ đề 4.2)</h2>

<table>

```

```

<tr>
    <th>ID</th>
    <th>Tên Sinh Viên</th>
    <th>Email</th>
    <th>Ngày Tạo</th>
</tr>
<?php
while ($row = $stmt_select->fetch(PDO::FETCH_ASSOC)) {
    echo "<tr>";
    echo "<td>" . htmlspecialchars($row['id']) . "</td>";
    echo "<td>" . htmlspecialchars($row['ten_sinh_vien']) . "</td>";
    echo "<td>" . htmlspecialchars($row['email']) . "</td>";
    echo "<td>" . htmlspecialchars($row['ngay_tao']) . "</td>";
    echo "</tr>";
}
?>
</table>
</body>
</html>

```

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 tệp chapter4.php mà bạn đã hoàn thiện.

B. Ảnh chụp màn hình Kết quả (BẮT BUỘC CẢ 2 ẢNH):

1. Ảnh 1 (phpMyAdmin): Chụp màn hình tab "Browse" (Duyệt) của bảng sinhvien trong phpMyAdmin, cho thấy bạn đã INSERT thành công ít nhất 2-3 sinh viên.
2. Ảnh 2 (Trình duyệt Web): Chụp ảnh màn hình trang chapter4.php của bạn, hiển thị đúng 2-3 sinh viên mà bạn vừa thêm (chứng minh SELECT thành công).

The screenshot shows the phpMyAdmin interface connected to a MySQL database named 'cse485_web'. The left sidebar lists databases like 'information_schema', 'mysql', 'performance_schema', and 'phpmyadmin'. The main area displays the 'sinhvien' table with 4 rows. The table has columns: id, ten_sinh_vien, email, and ngay_tao. The data is as follows:

ID	Tên Sinh Viên	Email	Ngày Tạo
1	Mã Khánh Linh	linhdeptra1@gmail.com	2025-11-29 21:41:08
2	Linh Đẹp Trai	linhdeptra1@gmail.com	2025-11-29 21:41:27
3	Linhdoncare	linhdeptra2@gmail.com	2025-11-29 21:41:40
4	Khánh Linh Mã	linhdeptra3@gmail.com	2025-11-29 21:41:52

The screenshot shows a web page at 'localhost/CNWeb/PHT/chapter4.php'. It features a form for adding a new student ('Thêm Sinh Viên Mới') and a table ('Danh Sách Sinh Viên') listing existing students. The table has columns: ID, Tên Sinh Viên, Email, and Ngày Tạo.

ID	Tên Sinh Viên	Email	Ngày Tạo
4	Khánh Linh Mã	linhdeptra3@gmail.com	2025-11-29 21:41:52
3	Linhdoncare	linhdeptra2@gmail.com	2025-11-29 21:41:40
2	Linh Đẹp Trai	linhdeptra1@gmail.com	2025-11-29 21:41:27
1	Mã Khánh Linh	linhdeptra1@gmail.com	2025-11-29 21:41:08

4. Câu hỏi Phản biến (Bắt buộc)

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

(Gợi ý: "Hãy giải thích SQL Injection là gì? Tại sao việc cộng chuỗi INSERT INTO sinhvien (ten) VALUES ('\$ten') lại nguy hiểm, và tại sao cách dùng execute(['\$ten']) (Prepared Statement) lại an toàn hơn?").

Câu hỏi của tôi là: Tại sao đã dùng PDO Prepared Statement để lưu dữ liệu an toàn vào CSDL rồi mà khi hiển thị dữ liệu đó ra màn hình (lệnh echo) vẫn phải dùng hàm htmlspecialchars()? Nếu không dùng thì bị lỗi gì?

5. Kết nối Đánh giá (Rất quan trọng)

Kỹ năng kết nối CSDL bằng PDO (bao gồm INSERT và SELECT) là kỹ năng quan trọng nhất trong khối kiến thức PHP thuần.

Bạn sẽ vận dụng trực tiếp PHT này để hoàn thành Bài tập trên lớp (Phần PHP), chiếm 20% tổng điểm, dự kiến vào Tuần 5. Năm vững PDO bây giờ cũng sẽ giúp bạn hiểu tại sao Eloquent ORM (Chương 8) lại mạnh mẽ và tiện lợi đến vậy.