

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 = '127.0.0.1'; // 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

// Gợi ý: Dùng isset($_POST['...']) if ( ... ) {

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

    $ten = ...;

    $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
// Gợi ý: $stmt = $pdo->prepare($sql);
// Gợi ý: $stmt->execute([$ten, $email]);

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

}

// === LOGIC LẤY DANH SÁCH SINH VIÊN (SELECT) ===
// TODO 7: Viết câu lệnh SQL SELECT * $sql_select = "SELECT * FROM
sinhvien ORDER BY ngay_tao DESC";
// TODO 8: Thực thi câu lệnh SELECT (không cần prepare vì không có tham số)
// Gợi ý: $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
            // TODO 9: Dùng vòng lặp (ví dụ: while) để duyệt qua kết quả $stmt_select
            // Gợi ý: while ($row = $stmt_select->fetch(PDO::FETCH_ASSOC)) { ... }

            // TODO 10: In (echo) các dòng <tr> và <td> chứa dữ liệu $row
            // Gợi ý: echo "<tr>";
            // Gợi ý: echo "<td>" . htmlspecialchars($row['id']) . "</td>";
```

// (htmlspecialchars là để bảo mật, tránh lỗi XSS - sẽ học ở Chương 9)

// Đóng vòng lặp

```
?>  
</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):

- 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.
- 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).

(Dán Code A

```
1. <?php  
2. // === THIẾT LẬP KẾT NỐI PDO ===  
3. $host = '127.0.0.1'; // hoặc localhost  
4. $dbname = 'cse485_web'; // Tên CSDL bạn vừa tạo  
5. $username = 'root'; // Username mặc định của XAMPP  
6. $password = ''; // Password mặc định của XAMPP (rỗng)  
7. $dsn = "mysql:host=$host;dbname=$dbname;charset=utf8mb4";  
8.  
9. try {  
10.    // TODO 1: Tạo đối tượng PDO để kết nối CSDL  
11.    $pdo = new PDO($dsn, $username, $password);  
12.  
13.    // Bật chế độ báo lỗi Exception để dễ debug  
14.    $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);  
15. } catch (PDOException $e) {  
16.    die("Kết nối thất bại: " . $e->getMessage());  
17. }  
18.  
19. // Biến thông báo (tùy chọn)  
20. $successMessage = '';  
21. $errorMessage = '';  
22.  
23. // === LOGIC THÊM SINH VIÊN (XỬ LÝ FORM POST) ===  
24. // TODO 2: Kiểm tra xem form đã được gửi đi (method POST) và có 'ten_sinh_vien' không  
25. if ($_SERVER['REQUEST_METHOD'] === 'POST' && isset($_POST['ten_sinh_vien'], $_POST['email'])) {  
26.  
27.    // TODO 3: Lấy dữ liệu 'ten_sinh_vien' và 'email' từ $_POST  
28.    $ten = trim($_POST['ten_sinh_vien']);  
29.    $email = trim($_POST['email']);  
30.  
31.    // Validate đơn giản (không bắt buộc nhưng nên có)
```

```

32.     if ($ten === '' || $email === '') {
33.         $errorMessage = "Vui lòng nhập đầy đủ tên và email.";
34.     } elseif (!filter_var($email, FILTER_VALIDATE_EMAIL)) {
35.         $errorMessage = "Email không hợp lệ.";
36.     } else {
37.         // TODO 4: Viết câu lệnh SQL INSERT với Prepared Statement (dùng dấu ?)
38.         $sql = "INSERT INTO sinhvien (ten_sinh_vien, email) VALUES (?, ?)";
39.
40.         try {
41.             // TODO 5: Chuẩn bị (prepare) và thực thi (execute) câu lệnh
42.             $stmt = $pdo->prepare($sql);
43.             $stmt->execute([$ten, $email]);
44.
45.             // TODO 6: Chuyển hướng về chính trang này để tránh submit lại khi F5
46.             header("Location: chapter4.php?success=1");
47.             exit;
48.         } catch (PDOException $e) {
49.             $errorMessage = "Lỗi khi thêm sinh viên: " . $e->getMessage();
50.         }
51.     }
52. }
53.
54. // Thông báo sau redirect
55. if (isset($_GET['success'])) && $_GET['success'] == '1') {
56.     $successMessage = "Thêm sinh viên thành công!";
57. }
58.
59. // === LOGIC LẤY DANH SÁCH SINH VIÊN (SELECT) ===
60. // TODO 7: Viết câu lệnh SQL SELECT *
61. $sql_select = "SELECT * FROM sinhvien ORDER BY ngay_tao DESC";
62.
63. try {
64.     // TODO 8: Thực thi câu lệnh SELECT (không cần prepare vì không có tham số)
65.     $stmt_select = $pdo->query($sql_select);
66. } catch (PDOException $e) {
67.     die("Lỗi SELECT: " . $e->getMessage());
68. }
69. ?>
70.
71. <!DOCTYPE html>
72. <html lang="vi">
73. <head>
74.     <meta charset="UTF-8">
75.     <title>PHT Chương 4 - Website hướng dẫn liệu</title>
76.     <style>
77.         body { font-family: Arial, sans-serif; margin: 24px; }
78.         h2 { margin-top: 28px; }
79.         .msg { padding: 10px 12px; border-radius: 8px; margin: 12px 0; }
80.         .success { background: #e8ffff; border: 1px solid #a7f3c0; color: #0f5132; }

81.         .error { background: #fff0f0; border: 1px solid #ffb3b3; color: #842029; }
82.         form { display: flex; gap: 10px; align-items: center; flex-wrap: wrap; }
83.         input { padding: 8px 10px; border: 1px solid #ccc; border-radius: 8px; }
84.         button { padding: 8px 14px; border: none; border-
85.             radius: 8px; cursor: pointer; }
86.             button:hover { opacity: 0.9; }
87.             table { width: 100%; border-collapse: collapse; margin-top: 12px; }
88.             th, td { border: 1px solid #ddd; padding: 10px; text-align: left; }
89.             th { background-color: #f2f2f2; }
90.         </style>
91.     </head>
92.     <body>
93.         <h2>Thêm Sinh Viên Mới (Chủ đề 4.3)</h2>
94.         <?php if ($successMessage): ?>
95.             <div class="msg success"><?php echo htmlspecialchars($successMessage); ?></d
96.             iv>
97.             <?php endif; ?>

```

```

98.      <?php if ($errorMessage): ?>
99.          <div class="msg error"><?php echo htmlspecialchars($errorMessage); ?></div>
100.         <?php endif; ?>
101.         <form action="chapter4.php" method="POST">
102.             Tên sinh viên:
103.             <input type="text" name="ten_sinh_vien" required>
104.             Email:
105.             <input type="email" name="email" required>
106.             <button type="submit">Thêm</button>
107.         </form>
108.         <h2>Danh Sách Sinh Viên (Chủ đề 4.2)</h2>
109.         <table>
110.             <tr>
111.                 <th>ID</th>
112.                 <th>Tên Sinh Viên</th>
113.                 <th>Email</th>
114.                 <th>Ngày Tạo</th>
115.             </tr>
116.             <tr>
117.                 <td>1</td>
118.                 <td>Đặng Quốc Đạt</td>
119.                 <td>dangquocdat@yahoo.comvn@gmail.com</td>
120.                 <td>2025-12-14 23:15:29</td>
121.             </tr>
122.             <tr>
123.                 <td>2</td>
124.                 <td>Hồ Xuân Dân</td>
125.                 <td>eto@gmail.com</td>
126.                 <td>2025-12-14 23:20:05</td>
127.             </tr>
128.             <tr>
129.                 <td>3</td>
130.                 <td>Nguyễn Xuân Bình</td>
131.                 <td>iwantudye@gmail.com</td>
132.                 <td>2025-12-14 23:20:36</td>
133.             </tr>
134.         </table>
135.     </body>
136. </html>

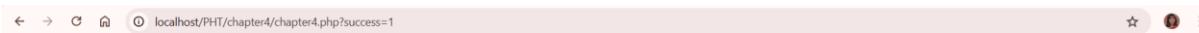
```

và Ảnh B1

The screenshot shows the MySQL Workbench interface with the 'sinhvien' table selected. The table has four columns: id, ten_sinh_vien, email, and ngay_tao. The data is as follows:

	ID	Tên Sinh Viên	Email	Ngày Tạo
1	1	Đặng Quốc Đạt	dangquocdat@yahoo.comvn@gmail.com	2025-12-14 23:15:29
2	2	Hồ Xuân Dân	eto@gmail.com	2025-12-14 23:20:05
3	3	Nguyễn Xuân Bình	iwantudye@gmail.com	2025-12-14 23:20:36

, B2 của bạn vào đây



Thêm Sinh Viên Mới

Thêm sinh viên thành công!

Tên sinh viên: Email: Thêm

Danh Sách Sinh Viên

ID	Tên Sinh Viên	Email	Ngày Tạo
3	Nguyễn Xuân Bình	iwantudye@gmail.com	2025-12-14 23:20:36
2	Hồ Xuân Dân	eto@gmail.com	2025-12-14 23:20:05
1	Đặng Quốc Đạt	dangquocdat@yahoo.comvn@gmail.com	2025-12-14 23:15:29

)

4. Câu hỏi Phản biện (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: (Vì sao Prepared Statement (prepare + execute) giúp chống SQL Injection tốt hơn so với việc nối chuỗi SQL trực tiếp từ dữ liệu người dùng nhập vào)

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 khái 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. Nắm vững PDO bây giờ cũng sẽ giúp bạn hiểu tại sao Eloquent ORM (Chương 8) lại mạnh mẽ và tiện lợi đến vậy.