Phiếu học tập chủ động

Môn học: CSE485- Công nghệ web

Họ và tên: Nguyễn Anh Tuấn

Mã sinh viên: 2251162199

Lớp: 64HTTT4

Lớp học phần: 65HTTT

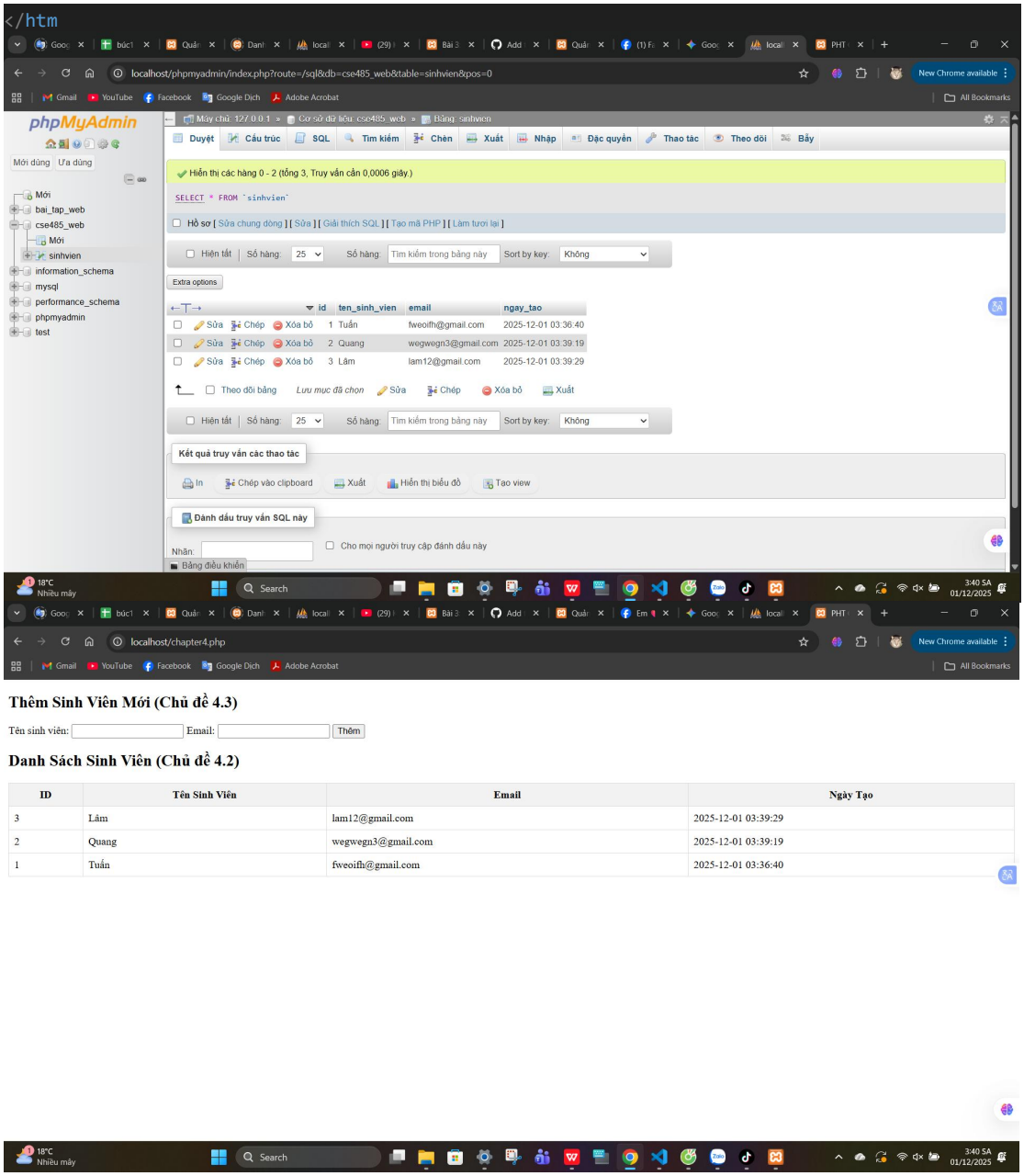
CHƯƠNG 4: Website hướng dữ liệu

Code:

- File chapter4.php:

```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 ($_SERVER['REQUEST_METHOD'] === 'POST' && isset($_POST['ten_sinh_vien'])) {

 // 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
 // Gợi ý: $stmt = $pdo->prepare($sql);
 // Gợi ý: $stmt->execute([$ten, $email]);
    $stmt = $pdo->prepare($sql);
    $stmt->execute([$ten, $email]);
 // TODO 6: (Tùy chọn) Chuyển hướng về chính trang này để "làm mới"
 // Gợi ý: Dùng header('Location: chapter4.php');
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);
$stmt_select = $pdo->query($sql_select);
?>
<!DOCTYPE html>
<html lang="vi">
<head>
 <meta charset="UTF-8">
 <title>PHT Chương 4 - Website hướng dữ 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
while ($row = $stmt_select->fetch(PDO::FETCH_ASSOC)) {
 // Gợi ý: while ($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

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>
```



Thêm Sinh Viên Mới (Chủ đề 4.3)

Tên sinh viên: Email: Thêm

Danh Sách Sinh Viên (Chủ đề 4.2)

ID	Tên Sinh Viên	Email	Ngày Tạo
3	Lâm	lam12@gmail.com	2025-12-01 03:39:29
2	Quang	wegwegn3@gmail.com	2025-12-01 03:39:19
1	Tuấn	fweoifh@gmail.com	2025-12-01 03:36:40

Câu hỏi phản biện:
Câu hỏi: Tại sao việc dùng Prepared Statement (dấu ?) lại an toàn hơn cộng chuỗi trực tiếp khi thực hiện câu lệnh SQL?
Câu trả lời:
SQL Injection là kỹ thuật tấn công nơi hacker chèn mã SQL độc hại vào đầu vào của người dùng.

Cách cộng chuỗi (Nguy hiểm): Nếu viết $sql = "SELECT ... WHERE user = '" . $_POST['user'] . "';, hacker có thể nhập ' OR '1'='1 để đánh lừa hệ thống và đăng nhập không cần mật khẩu.

Prepared Statement (An toàn): Khi dùng prepare với dấu ? (placeholder), cơ sở dữ liệu sẽ coi mọi dữ liệu người dùng nhập vào chỉ là văn bản thuần túy (data), không phải là mã lệnh (code). Do đó, dù hacker có nhập ký tự đặc biệt nào thì câu lệnh SQL cũng không bị thay đổi cấu trúc.