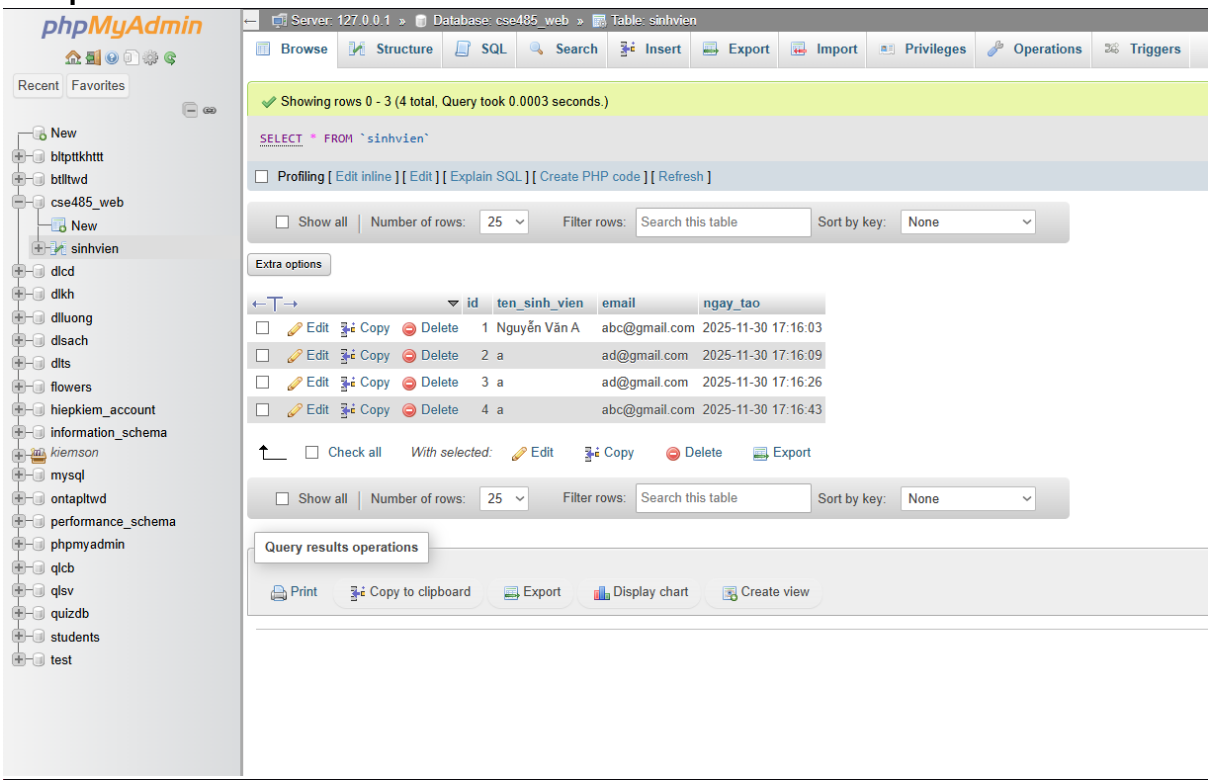
A. Code đã hoàn thiện

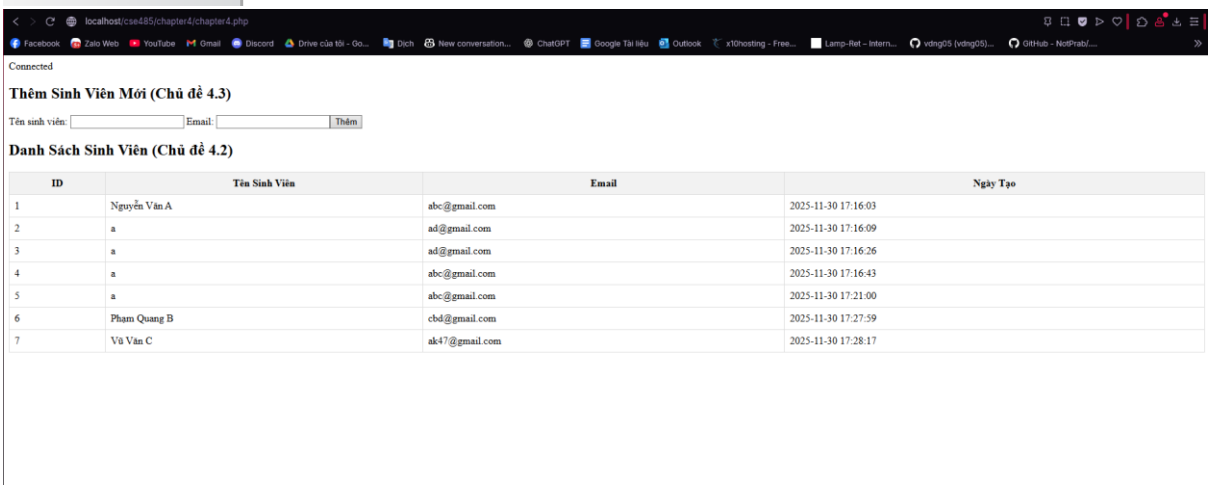
chapter4.php

```php
<?php
    $host = "127.0.0.1";
    $dbname = "cse485_web";
    $username = "root";
    $password = "";
    $dsn = "mysql:host=$host;dbname=$dbname;charset=utf8mb4";
    try
    {
        $pdo = new PDO(dsn: $dsn, username: $username, password: $password);
        $pdo->setAttribute(attribute: PDO::ATTR_ERRMODE, value: PDO::ERRMODE_EXCEPTION);
        echo "Connected";
    }
    catch (PDOException $e)
    {
        die("Connection failed: " . $e->getMessage());
    }

    if (isset($_POST["ten_sinh_vien"]) && isset($_POST["email"]))
    {
        $ten = $_POST["ten_sinh_vien"];
        $email = $_POST["email"];
        $sql = "INSERT INTO sinhvien (ten_sinh_vien, email) VALUES (?, ?)";
        $stmt = $pdo->prepare(query: $sql);
        $stmt->execute(params: [$ten, $email]);
        // echo "Added " . $ten . " with email " . $email;
        header(header: "Location: chapter4.php");
        exit;
    }
    $sql_select = "SELECT * FROM sinhvien";
    $stmt_select = $pdo->query(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
        // Gợi ý: while ($row = $stmt_select->fetch(PDO::FETCH_ASSOC)) { ... }
            while ($row = $stmt_select->fetch(mode: PDO::FETCH_ASSOC))
            {
                echo "<tr>";
                echo "<td>" . htmlspecialchars(string: $row['id']) . "</td>";
                echo "<td>" . htmlspecialchars(string: $row['ten_sinh_vien']) . "</td>";
                echo "<td>" . htmlspecialchars(string: $row['email']) . "</td>";
                echo "<td>" . htmlspecialchars(string: $row['ngay_tao']) . "</td>";
                echo "</tr>";
            }
        // 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>
```

B. Kết quả





C. Câu hỏi phản biện

- Tại sao Prepared Statement với PDO lại có thể ngăn chặn hoàn toàn SQL Injection trong khi việc ghép chuỗi trực tiếp (string concatenation) lại cực kỳ nguy hiểm?