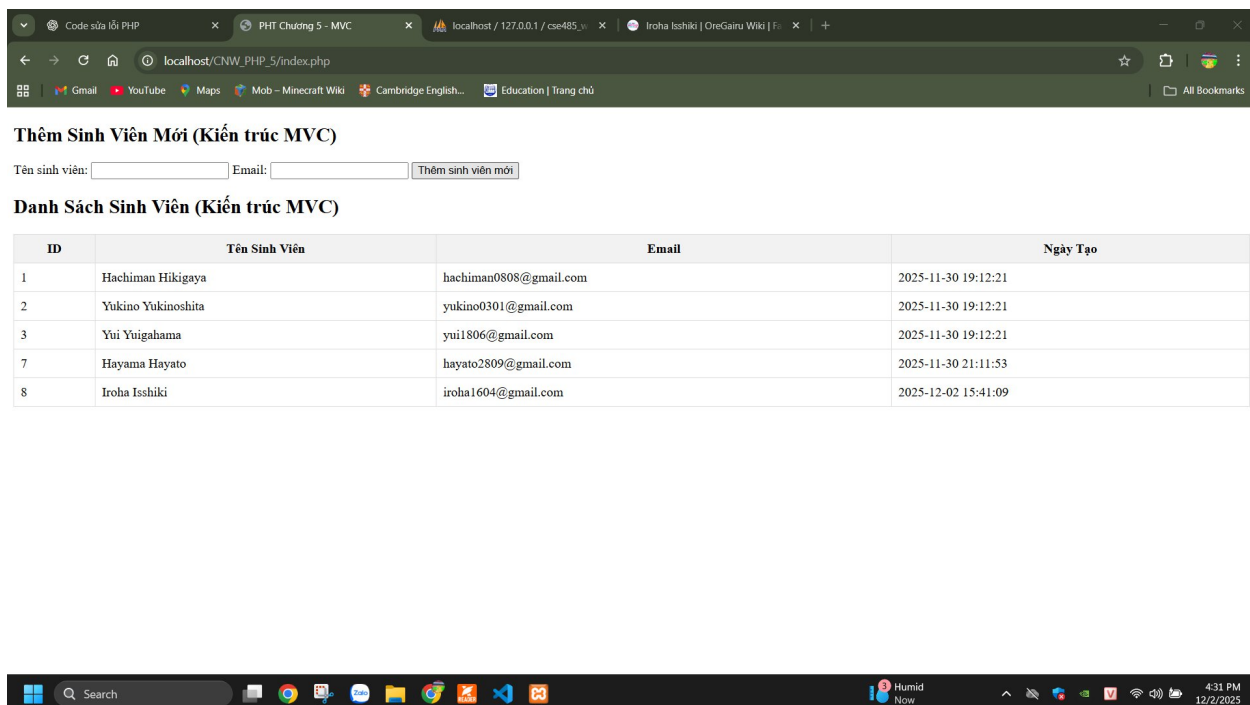


PHIẾU BÀI TẬP CHƯƠNG 5

1. Ảnh chụp màn hình kết quả



2. Câu hỏi tư duy (Sinh viên thử tìm câu trả lời)

MVC = Model - View - Controller

- **Model:** chứa **logic dữ liệu**, thao tác CSDL, truy vấn, insert, update...
- **View:** chứa **HTML + hiển thị dữ liệu**, KHÔNG thao tác trực tiếp với database
- **Controller:** điều phối, gọi Model lấy dữ liệu, gửi sang View

Câu 1: MVC là tưởng tượng giống như một mô hình viết code để chia logic cho gọn file nhất có thể bởi vì qua bài tập này tôi thấy nó chỉ là nhúng code qua include với require giống như cố gắng lắp chơi ghép hình. Vậy nghĩa là bản chất nếu viết chung 1 file cũng không khác nhau là mấy?

Trả lời: Về kỹ thuật, không khác nhau nhiều: PHP vẫn chạy tuần tự, vẫn xử lý logic và echo HTML. Nhưng về quản lý code và mở rộng:

- Khi viết tất cả trong 1 file:
 - Logic DB, HTML, request handling trộn lẫn → khó bảo trì
 - Muốn thay đổi logic hoặc hiển thị → dễ phá hỏng phần khác

- Khó tái sử dụng code, test từng phần riêng biệt
- Khi dùng MVC:
 - Tách Model/Controller/View → rõ ràng
 - Muốn thay đổi HTML → sửa View, không động đến Model
 - Muốn thay DB hoặc logic → sửa Model, View vẫn chạy được
 - Dễ mở rộng project lớn

Câu 2: Trong file `views/sinhvien_view.php`, tại sao chúng ta tuyệt đối không được phép viết code kết nối PDO hay `SELECT`? Lợi ích của việc 'cấm' View làm việc trực tiếp với CSDL là gì?

Trả lời:

Lý do:

- View chỉ để hiển thị, không cần biết dữ liệu từ đâu. Nếu View truy cập CSDL trực tiếp, View vừa hiển thị vừa thao tác DB (khó debug, khó tái sử dụng).
- Nếu View tự `SELECT`, mỗi view khác muốn dữ liệu lại phải viết lại truy vấn. Càng nhiều view (càng khó bảo trì, khó sửa đổi truy vấn).
- Thao tác database trực tiếp trong View, dễ lộ thông tin nhạy cảm, dễ viết SQL kém an toàn (SQL injection). Nếu mọi dữ liệu đều đi qua Model (để kiểm soát, validate, sanitize).
- Muốn thay đổi DB (MySQL sang PostgreSQL), hoặc đổi logic lọc dữ liệu, phải sửa từng view. Nếu có Model xử lý chỉ sửa Model, View không cần động tay.

Lợi ích:

- Dễ bảo trì: Model xử lý dữ liệu, View hiển thị → tách rạch ròi
- Dễ mở rộng: Thêm view mới chỉ cần gọi dữ liệu từ Controller/Model
- An toàn hơn: Không expose logic DB trong HTML
- Tái sử dụng code: Model dùng chung cho nhiều controller/view