* **Định nghĩa:**

MVC là mô hình kiến trúc phần mềm chia ứng dụng thành 3 thành phần:

+ Model: quản lí dư liệu và logic.

+ View: Hiển thị giao diện người dùng.

+ Controller: Điều phối giữa Model và View.

* **Cách thức hoạt động:**

Người dùng gửi yêu cầu (Request):

* Ví dụ: Người dùng nhấn nút "Xem danh sách sản phẩm".
* Yêu cầu này được gửi đến Controller.

Controller xử lý yêu cầu:

* Controller sẽ phân tích yêu cầu và gọi các phương thức tương ứng trong Model.
* Ví dụ: ProductController gọi ProductModel::getAll() để lấy danh sách sản phẩm.

Model xử lý dữ liệu:

* Model thực hiện truy vấn cơ sở dữ liệu và xử lý nghiệp vụ.
* Dữ liệu sẽ được trả về cho Controller.

Controller chuyển dữ liệu sang View:

* Controller nhận dữ liệu từ Model, chuẩn bị và truyền dữ liệu cho View.

View hiển thị dữ liệu cho người dùng:

* View nhận dữ liệu và hiển thị nó dưới dạng HTML (hoặc giao diện người dùng).
* Ví dụ: Một bảng danh sách sản phẩm được hiển thị.

Người dùng tiếp tục tương tác → quay lại bước 1.

* **Ví dụ:**

+ Bạn là người dùng (user).

+ Người phục vụ là Controller.

+ Quầy pha chế là Model.

+ Ly cà phê bưng ra bàn là View.

+ Quá trình:

1. Gọi món = gửi yêu cầu(request)
2. Người phục vụ tiếp nhận yêu cầu → Controller xử lý
3. Quầy pha chế làm đồ uống → Model xử lý dữ liệu
4. Người phục vụ nhận ly cà phê và mang ra bàn → View hiển thị
5. Bạn thưởng thức cà phê → Nhận phản hồi

* **Ưu và nhược của MVC:**

1. Ưu điểm:

+ Làm gì cũng có người phụ trách riêng → rõ ràng, gọn gàng, không ai làm việc của ai. Thay đổi giao diện thì không ảnh hưởng dữ liệu. Rất dễ sửa và nâng cấp.

+ Dễ sửa, dễ nâng cấp. Không phải “đụng tới cả đống code” khi muốn thay đổi một thứ nhỏ.

+ Code viết ra có thể tái sử dụng nhiều chỗ.

+ Làm nhóm dễ chia việc.

+ Tốt cho các dự án lớn, chuyên nghiệp.

1. Nhược điểm:

+ Không phù hợp cho dự án nhỏ.

+ Khó học.

+ Nếu tổ chức không tốt sẽ bị rối vì quản lí nhiều file.

+ Thiết lập ban đầu phức tạp.