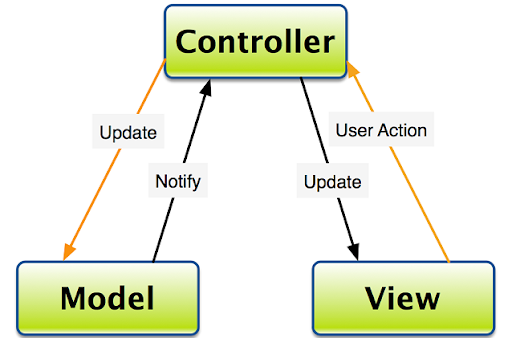
**Giải thích Mô hình M - V - C**

****

**Mô hình M-V-C:** là một mô hình lập trình nhằm tách biệt các module trong chương trình thành các nhóm mã lệnh độc lập, thực thi các tác vụ chuyên biệt không phụ thuộc vào các mã lệnh và các công nghệ khác. Nhằm chuyên môn hóa các module và cho phép chia việc trong nhóm lập trình theo khả năng của mỗi thành viên, cho phép dễ dàng tái sử dụng các module đưa vào dự án khác, dễ dàng nâng cấp và sửa lỗi mà không ảnh hưởng đến các module khác.

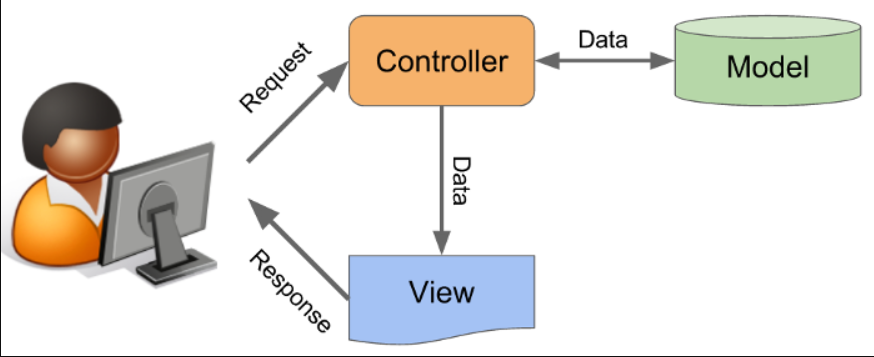
**Ví dụ :**

Nhóm Model: chỉ chứa các mã lênh PHP (java) làm việc với CSDL và không phụ thuộc (không liên quan đến công nghệ HTML, CSS… hay mã lệnh Window Form, Mobile…), việc này cho phép các mã lệnh trong Model được copy sang dùng lại ở các dự án khác mà không phụ thuộc vào loại dự án là Web, Winform…

Mô hình M-V-C cho phép chia công việc trong 1 nhóm lập trình cùng dự án, sao cho mỗi thành viên thạo về lĩnh vực thì chỉ tập trung vào lĩnh vực đó. Thành viên giỏi SQL thì code phần Model (không liên quan đến giao diện). Thành viên giỏi về giao diện (HTML, CSS, JASCRIPT…) thì làm phần View để trình diễn, chỉ cần sử dụng Controller đẻ tạo và gọi hàm của Model và sử dụng các biến kết quả để hiển thị.

* **Model**: Chỉ gồm các lớp, hàm thực hiện việc kết nối và làm việc với CSDL (để thực hiện các công việc tìm kiếm dữ liệu, cập nhật dữ liệu…) mà không có các mã lệnh hiển thị giao diện người dùng. Đầu ra của các hàm và các phương thức của các lớp chỉ là các biến, đối tượng, mảng… chứa dữ liệu
* **View**: Gồm các lớp, hàm, tệp php… để hiện thị giao diện người dùng và nội dung các biến (biến đơn, biến mảng, biến kiểu đối tượng…), mà các biến này được được tạo ra và cung cấp từ các controller tương ứng trước khi gọi View
* **Controller**: Là các lớp, hàm tệp PHP điều khiển chương trình. Người dùng khi truy cập vào ứng dụng thì sẽ truy cập vào các controller cần thực thi

**Qui trình thực hiện:**



Các bước thực thiện cơ bản:

B1. Người dùng truy cập vào Controller tương ứng với công việc cần thực hiện

B2. Controller sẽ nhận các yêu cầu (Request) từ người dùng => gọi các đối tượng, hàm từ **Model** để lấy các dữ liệu liên quan đến chức năng (act) cần thực hiện, lưu vào các biến dùng chung

B3. Controller sẽ gọi các View hoặc Controller thứ cấp (cấp dưới chi tiết hơn) để thực thi và các View và Controller thứ cấp sẽ sử dụng các dữ liệu (từ các biến, đối tượng…) để hiển thị và thực hiện các công việc liên quan => Hiển thị giao diện và kết quả lên màn hình của người dùng