**MVC** là viết tắt của cụm từ “**Model-View-Controller**“. Đây là mô hình thiết kế được sử dụng trong kỹ thuật phần mềm. MVC là một **mẫu kiến trúc phần mềm** để tạo lập giao diện người dùng trên máy tính. MVC chia thành ba phần được kết nối với nhau và mỗi thành phần đều có một **nhiệm vụ riêng** của nó và **độc lập** với các thành phần khác. Tên gọi 3 thành phần:

* **Model (dữ liệu)**: Quản lí xử lí các dữ liệu.
* **View (giao diện)**: Nới hiển thị dữ liệu cho người dùng.
* **Controller (bộ điều khiển)**: Điều khiển sự tương tác của hai thành phần **Model** và **View**

**MVC** được tiến sĩ Trygve Reenskaug đưa vào [ngôn ngữ lập trình](https://vietnix.vn/ngon-ngu-lap-trinh/) Smalltalk-76 khi ông đến trung tâm Nghiên cứu Xerox Palo Alto (PARC) vào giữa năm 1970. Sau đó, việc triển khai trở nên phổ biến trong các phiên bản khác của Small- Talk. Năm 1988, các bài báo “The Journal of Object Technology” – JOT mang lại bước tranh toàn cảnh về MVC mang liệu sự hiệu quả tốt nhất.

Luồng xử lý trong của mô hình MVC, bạn có thể hình dung cụ thể và chi tiết qua từng bước dưới đây:

* Khi một yêu cầu của từ máy khách (Client) gửi đến Server. Thì bị Controller trong MVC chặn lại để xem đó là URL request hay sự kiện.
* Sau đó, **Controller** xử lý **input** của user rồi giao tiếp với**Model** trong MVC.
* Model chuẩn bị data và gửi lại cho Controller.
* Cuối cùng, khi xử lý xong yêu cầu thì Controller gửi dữ liệu trở lại View và hiển thị cho người dùng trên trình duyệt.

Ở đây, **View không giao tiếp trực tiếp với Model**. Sự tương tác giữa **View và Model**sẽ chỉ được xử lý bởi **Controller.**

## Ưu và nhược điểm của MVC

### Ưu điểm mô hình MVC

* Đầu tiên, nhắc tới ưu điểm mô hình MVC thì đó là băng thông ([Bandwidth](https://vietnix.vn/bandwidth-la-gi/)) nhẹ vì không sử dụng viewstate nên khá tiết kiệm băng thông. Việc giảm băng thông giúp website hoạt động ổn định hơn.
* Kiểm tra đơn giản và dễ dàng, kiểm tra lỗi phần mềm trước khi bàn giao lại cho người dùng.
* Một lợi thế chính của MVC là nó tách biệt các phần Model, Controller và View với nhau.
* Sử dụng mô hình MVC chức năng Controller có vai trò quan trọng và tối ưu trên các nền tảng ngôn ngữ khác nhau
* Ta có thể dễ dàng duy trì ứng dụng vì chúng được tách biệt với nhau.
* Có thể chia nhiều developer làm việc cùng một lúc. Công việc của các developer sẽ không ảnh hưởng đến nhau.
* Hỗ trợ [TTD](https://en.wikipedia.org/wiki/Test-driven_development) (test-driven development). Chúng ta có thể tạo một ứng dụng với unit test và viết các won [test case](https://vietnix.vn/test-case-la-gi/).
* Phiên bản mới nhất của MVC hỗ trợ trợ thiết kế responsive website mặc định và các mẫu cho mobile. Chúng ta có thể tạo công cụ View của riêng mình với cú pháp đơn giản hơn nhiều so với công cụ truyền thống.

### Nhược điểm mô hình MVC

Bên cạnh những ưu điểm MVC mang lại thì nó cũng có một số nhược điểm cần khắc phục.

MVC đa phần phù hợp với công ty chuyên về website hoặc các dự án lớn thì mô hình này phù hợp hơn so với với các dự án nhỏ, lẻ vì khá là cồng kềnh và mất thời gian.

## Vì sao nên sử dụng mô hình MVC?

### **Quy trình phát triển nhanh hơn**

**MVC** hỗ trợ phát việc phát triển nhanh chóng và song song. Nếu một **mô hình MVC** được dùng để phát triển bất kỳ ứng dụng web cụ thể nào, một lập trình viên có thể làm việc trên **View** và một developer khác có thể làm việc với **Controller** để tạo logic nghiệp vụ cho ứng dụng web đó.

Do đó, **ứng dụng mô hình MVC** có thể được hoàn thành nhanh hơn ba lần so với các ứng dụng mô hình khác.

### **Khả năng cung cấp nhiều chế độ view**

Trong **mô hình MVC**, bạn có thể tạo nhiều View cho **chỉ một mô hình**. Ngày nay, nhu cầu có thêm nhiều cách mới để truy cập ứng dụng và đang ngày càng tăng. Do đó, việc sử dụng MVC để phát triển chắc chắn là một giải pháp tuyệt vời.

Hơn nữa, với phương pháp này, việc nhân bản code rất hạn chế. Vì nó tách biệt dữ liệu và logic nghiệp vụ khỏi màn hình.

### **Các sửa đổi không ảnh hưởng đến toàn bộ mô hình**

Đối với bất kỳ ứng dụng web nào, người dùng có **xu hướng thay đổi** thường xuyên. Bạn có thể quan sát thông qua những thay đổi thường xuyên về màu sắc, font chữ, bố cục màn hình. Hay là thêm hỗ trợ thiết bị mới cho điện thoại hay máy tính bảng…

Việc thêm một kiểu view mới trong MVC rất đơn giản. Vì phần Model không phụ thuộc vào phần View. Do đó, bất kỳ thay đổi nào trong Model sẽ không ảnh hưởng đến toàn bộ kiến trúc.

### **MVC Model trả về dữ liệu mà không cần định dạng**

**MVC pattern** có thể trả về dữ liệu mà không cần áp dụng bất kỳ định dạng nào. Do đó, các thành phần giống nhau có thể được sử dụng với bất kỳ giao diện nào.

**Ví dụ:** tất cả loại dữ liệu đều có thể được định dạng bằng HTML. Ngoài ra, nó cũng có thể được định dạng bằng **Macromedia Flash** hay **Dream Viewer.**