

**1. Định nghĩa Mô hình MVC**

MVC (Model-View-Controller) là một kiến trúc phần mềm được sử dụng phổ biến trong phát triển ứng dụng web. Nó chia ứng dụng thành ba thành phần chính: Model, View, và Controller. Mục tiêu của MVC là tách biệt các phần khác nhau của ứng dụng, làm cho việc phát triển, bảo trì và mở rộng ứng dụng dễ dàng hơn.

* Model: Quản lý dữ liệu và logic nghiệp vụ của ứng dụng.
* View: Chịu trách nhiệm hiển thị dữ liệu và giao diện người dùng.
* Controller: Điều khiển luồng của ứng dụng, xử lý yêu cầu từ người dùng và tương tác với Model và View để trả về phản hồi phù hợp.

**2. Các thành phần của MVC**

**Model**

*Định nghĩa*: Model đại diện cho dữ liệu và các quy tắc nghiệp vụ của ứng dụng. Nó chịu trách nhiệm truy xuất, lưu trữ, và xử lý dữ liệu từ cơ sở dữ liệu hoặc các nguồn dữ liệu khác.

*Nhiệm vụ:*

* Quản lý dữ liệu: CRUD (Create, Read, Update, Delete).
* Ánh xạ dữ liệu với các đối tượng trong ứng dụng.
* Kiểm tra tính hợp lệ của dữ liệu.
* Thực hiện các phép toán và logic nghiệp vụ.

*Ví dụ:*

Một lớp Product trong ứng dụng bán hàng, chứa các thuộc tính như name, price, và các phương thức để tương tác với cơ sở dữ liệu như save(), delete().

**View**

*Định nghĩa:* View chịu trách nhiệm hiển thị dữ liệu cho người dùng và xử lý giao diện người dùng (UI). View lấy dữ liệu từ Model và hiển thị nó theo cách mà người dùng có thể tương tác.

*Nhiệm vụ:*

* Hiển thị thông tin từ Model cho người dùng.
* Tạo giao diện người dùng (UI) như các trang HTML, template, hoặc kết quả JSON/XML.
* Không chứa logic nghiệp vụ, chỉ tập trung vào việc hiển thị.

*Ví dụ:*

Một file product\_list.php hiển thị danh sách sản phẩm từ Model dưới dạng bảng HTML.

**Controller**

*Định nghĩa:* Controller là cầu nối giữa Model và View. Nó nhận yêu cầu từ người dùng, xử lý các yêu cầu này bằng cách tương tác với Model, và quyết định View nào sẽ được trả về để hiển thị kết quả cho người dùng.

*Nhiệm vụ:*

* Nhận và xử lý các yêu cầu HTTP từ người dùng.
* Tương tác với Model để xử lý dữ liệu hoặc thực hiện các hành động nghiệp vụ.
* Chọn View phù hợp để trả về kết quả cho người dùng.

*Ví dụ:*

Một lớp ProductController có phương thức index() để lấy danh sách sản phẩm từ Model và chuyển chúng đến View để hiển thị.

**3. Ưu và Nhược điểm của MVC**

**Ưu điểm:**

*Tách biệt rõ ràng giữa các phần của ứng dụng:*

* Giúp việc phát triển, bảo trì, và mở rộng ứng dụng dễ dàng hơn.
* Mỗi phần (Model, View, Controller) có thể được phát triển và kiểm thử riêng biệt.

*Tái sử dụng mã:*

* Các thành phần trong Model có thể được tái sử dụng ở nhiều nơi khác nhau trong ứng dụng.
* View có thể được thay đổi mà không cần thay đổi logic nghiệp vụ (Model).

*Quản lý mã nguồn dễ dàng:* Dễ dàng quản lý mã nguồn trong các dự án lớn với nhiều nhà phát triển làm việc cùng nhau.

*Khả năng mở rộng:* Ứng dụng có thể dễ dàng mở rộng thêm chức năng mà không làm gián đoạn cấu trúc hiện tại.

**Nhược điểm:**

* Phức tạp hơn trong các ứng dụng nhỏ: MVC có thể làm cho các ứng dụng nhỏ trở nên phức tạp hơn so với cần thiết, do phải chia mã nguồn thành nhiều thành phần riêng biệt.
* Tăng số lượng file và cấu trúc phức tạp: Do MVC chia ứng dụng thành nhiều thành phần, số lượng file trong ứng dụng sẽ tăng lên, làm cho việc quản lý dự án trở nên phức tạp hơn.
* Đòi hỏi kinh nghiệm và kỹ năng tốt: Để sử dụng MVC một cách hiệu quả, nhà phát triển cần phải có hiểu biết sâu về kiến trúc phần mềm và các nguyên tắc thiết kế.