

SỔ TAY LÀM WEB

BUỔI 1

- Yêu cầu 3.1:

Tạo mẫu tin ánh xạ giữa tên miền và địa chỉ IP

BUỔI 2:

Các tầng của webserver :

- Đặt website lên server
- Phần mềm webserver(apache,IIS,database management system....)
- Hệ điều hành (server)
- Phần cứng

Port (cổng): xác định chính xác 1 tab để mở gói dữ liệu ra (vd:mail,web....)

6 loại Hosting

- Shared Hosting
- VPS Hosting
- Cloud Hosting
- WordPress Hosting
- Dedicated Server Hosting

BUỔI 3

Phần mềm quản lý phiên bản (version control "VCS"): SVN(tập trung) - GIT(PHÂN TÁN).

Github(remote git), gitlab,bitbucket : Các con của Git.

Git: khi người dùng lấy phiên bản về thì sẽ lấy tất cả các phiên bản về 1 lúc, khi máy chủ chết ko bị mất dữ liệu.1 tuần có thể đẩy dữ liệu lên 1 lần

tạo 1 thư mục trong D\project

đến git \$cd d:

\$cd project

\$git init (initalized)

git commit (lưu tập tin đã được stage vào thư mục .git)

BUỔI 4

HTML đơn giản:(HyperText Markup Language).

Ngôn ngữ đánh dấu để tạo ra siêu văn bản.

Siêu văn bản là: text+hình ảnh+video.

Siêu văn bản:dùng văn bản để mô tả văn bản.

<h1>Chương 1</h1> header level 1.

Cấu trúc (tài liệu) + ngữ nghĩa (cho tài liệu HTML).

div (division): bộ phận.

a (anchor): mỏ neo.:
relative path: không bắt đầu từ gốc.
absolute path: xuất phát từ gốc(web root /).

BUỔI 5:

Viết css là viết các luật

selector {property :value} => bộ chọn{thuộc tính: giá trị}

css:cascading style sheet.

- để mã css ở đâu:
 - inline (cùng hàng với html)-- ít dùng.
 - embedded (đặt trong phần tử style).
 - external (mở rộng ra bên ngoài).

conflict: Xung đột.

+ Độ ưu tiên của Css

CMS:

- CMS (content management system): hệ quản trị nội dung(wordpress,drupal,joomla,)
- Framework (khung): Asp.net, laravel,Django....
- Viết thuần: HTML, CSS, java,python,C++,....

JAVASCRIPT 1.

- scripting language: ngôn ngữ kịch bản (là ngôn ngữ thông dịch "interpret")
- Interpreter: PHP JavaScript
- Ngôn ngữ biên dịch (compile - compiler): C, C++, C#
- JavaScript thuần == ECMAScript(ES6,ES7....)
- SPA: single page application

JAVASCRIPT 2.

- 2 loại ngôn ngữ lập trình:
 - Định kiểu mạnh: int x, int y.
 - Định kiểu yếu:var x ={ }

BOOTSTRAP.

HTML: CẤU trúc nội dung, tạo ngữ nghĩa cho tài liệu.

CSS: trang trí trang web.

Javascript: thực hiện các xử lý động (jQuery)

Framework: (bootstrap) tạo giao diện nhanh,tương thích.

Front- end (vue,react...)

php, java, C#, Python, Ruby, javascript. Nhưng javascript khác những cái trước

PHP và javascript: (server, back-end).

4 hướng tiếp cận để làm phần mềm (Lập trình)

- Hướng cấu trúc: main(), tạo ra hệ thống menu.
- Hướng đối tượng: Xây dựng các đối tượng(dữ liệu, phương thức)
- Hướng sự kiện: main (form), xây dựng các control, xử lý các event (winform)
- Hướng dịch vụ: gọi các dịch vụ (wordpress,plugin)

Hướng đối tượng:

- Tạo đối tượng: thuộc tính, phương thức
- Tính đóng gói (encapsulation): private, protected, public
- Kế thừa: (inheritance)
- Tính đa hình (polymorphism)

Hỏi 3 mức độ

- Biến (gì? what?)
- Hiểu (why?)
- làm được (how)

Chân thành, khiêm tốn.

Mini Portal

Agile/Scrum : Linh hoạt/ cộng tác

Prototyping

Quy trình làm dự án:

- Lấy dự án
- Viết bản Tôn chỉ dự án (Lưu hành nội bộ)
- Dựa vào các kết quả chuyển giao để viết WBS (Bảng phân rã công việc)
- Đưa lên trello
- Nhóm bắt đầu công việc

MySQL

Database

```
struct sv {
```

```
    tên: Tèo,  
    tuổi: 20,  
    lớp: cntt
```

```
}
```

filewrite (dulieu.txt); >> cơ sở dữ liệu dựa trên tập tin (.xls, .csv, .txt)

tên: Tèo Tuổi: 18 lớp: CNTT mssv: 1234

fileread (dulieu.txt);

Table sv

mssv		tên		tuổi		lớp	
1234		Tèo		18		cntt	

Table top

MERN

mongodb,json: cơ sở dữ liệu bán cấu trúc

```
sv {  
  tên : Tèo,  
  tuổi: 18,  
  mssv: 1234,  
  lớp: {  
    tên: cntt,  
    sl: 40  
  }  
}
```

Cơ sở dữ liệu hướng đối tượng

```
sv {  
  tên : Tèo,  
  tuổi: 18,  
  mssv: 1234,  
  lớp: {  
    tên: cntt,  
    sl: 40  
  }  
  methods(): { }  
}
```

1. Tìm hiểu cách tạo cơ sở dữ liệu
2. Cách truy vấn để lấy ra dữ liệu theo yêu cầu
3. Phân quyền, bảo mật, viết hàm truy vấn
4. Sử dụng trong các ngôn ngữ lập trình (kết nối, CRUD)
5. MVC, CMS

Framework laravel

PHP khung làm việc (design pattern=OOP nâng cao)

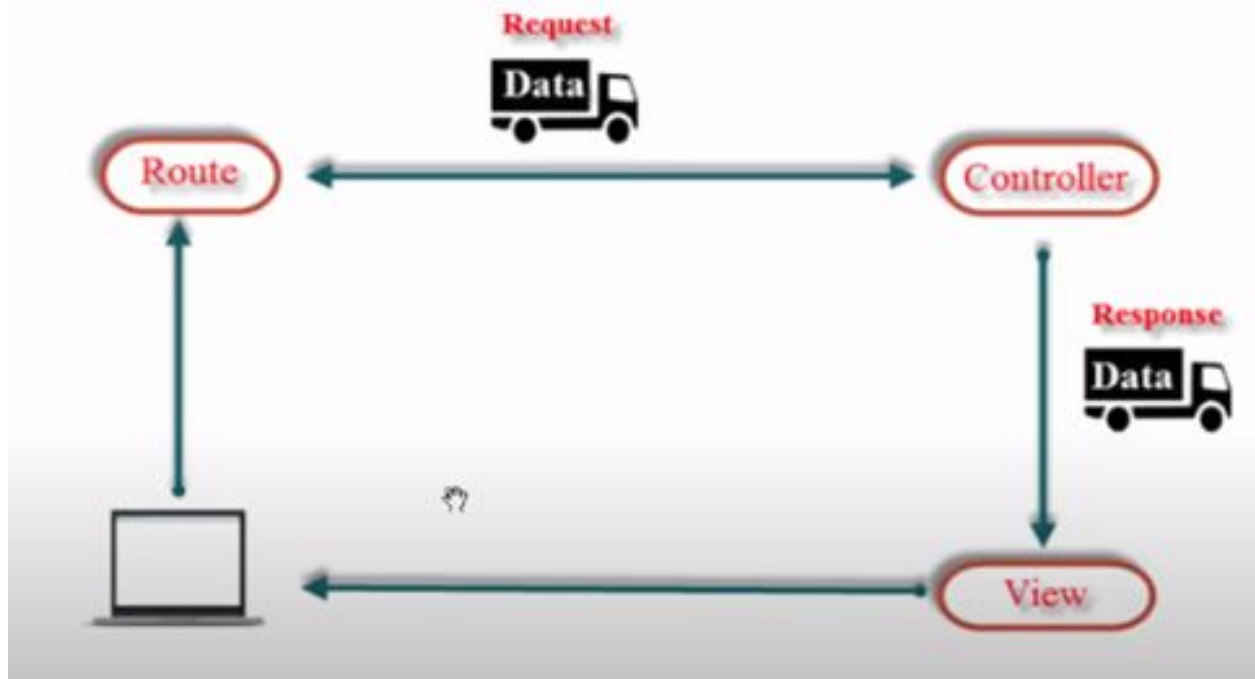
Cách làm việc MVC:

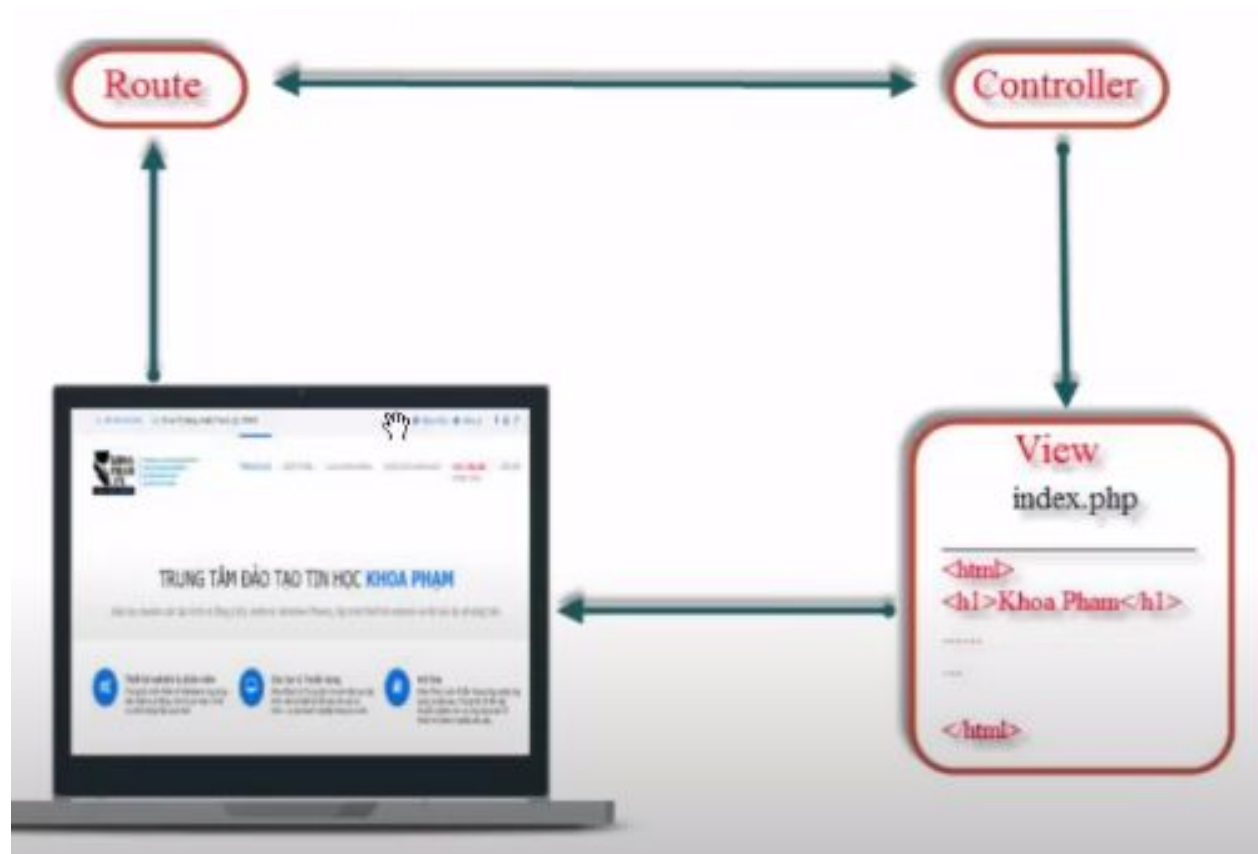
- Đường dẫn từ trình duyệt sẽ đc chuyển qua tập tin route

- route sẽ gọi hàm tương ứng trong controller
- controller sẽ xử lý: đọc dữ liệu từ model, gửi dữ liệu qua View
- cơ chế chuyển dữ liệu giữa Model - Controller - view

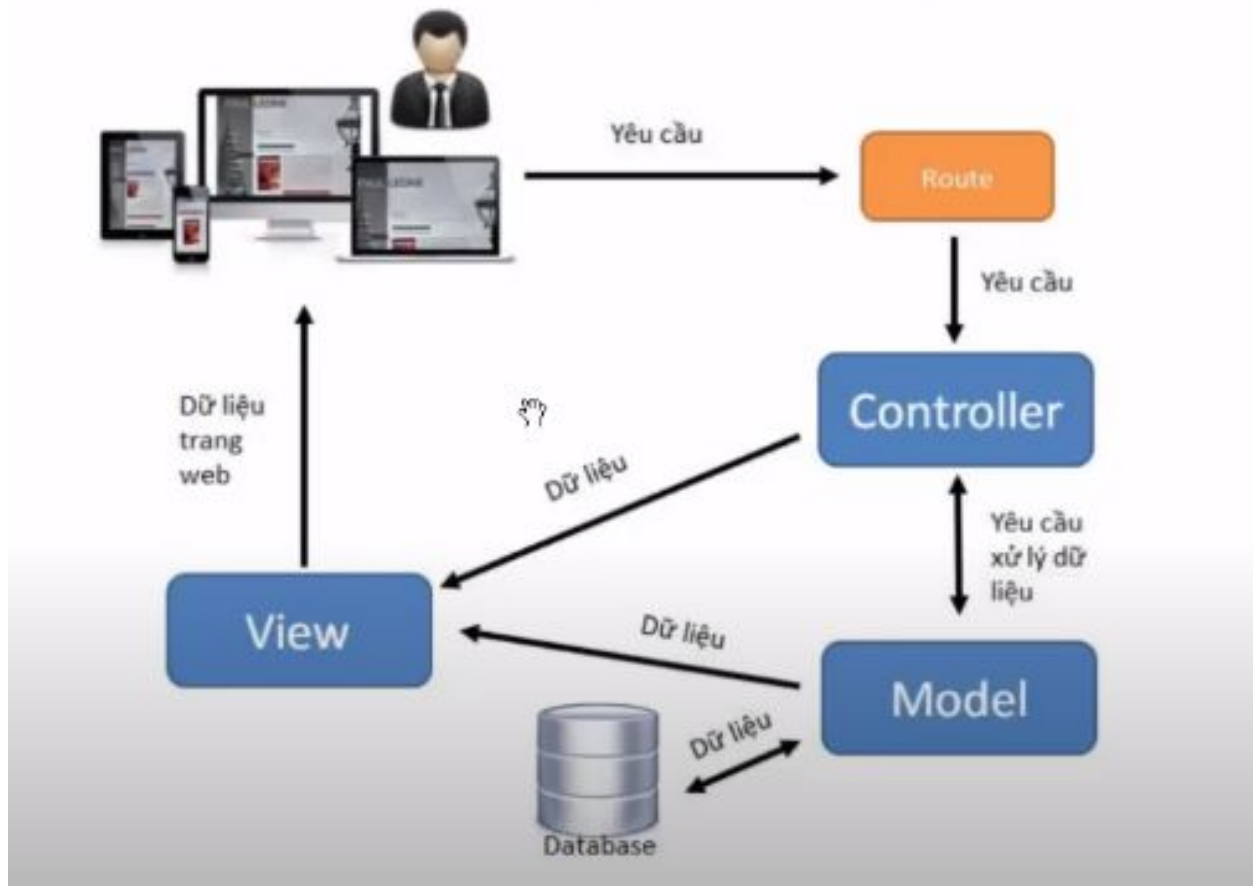
Tạo project mới: `composer create-project --prefer-dist laravel/laravel MyLaravel`.

Tạo controller: `php artisan make:controller MyController`.





Mô hình : Model View Controller



- Controller: đảm nhiệm công việc sắp xếp xử lý các công việc yêu cầu của người dùng.
- Model: đảm nhiệm trao đổi dữ liệu với csdl và gửi lại cho Controller và View.
- View: file giao diện giúp hiển thị dữ liệu lên cho người dùng xem

CMS

CMS: content management system (hệ thống quản lý nội dung web)

- Thuần
- framework
- cms

Framework trên java (demandware)

Phát triển phần mềm theo mô hình lặp (iterative, prototyping)

- lấy yêu cầu của khách (sơ bộ)
- làm một website mẫu > mang tới hỏi khách hàng > chỉnh sửa liên tục

MERN

MERN stack : MongoDB, Express, React, Node, web, mobile, desktop, nhúng.
Dựa trên JS làm nền.

mongodb+srv://user1:<password>@cluster0-flndp.mongodb.net/<dbname>?retryWrites=true&w=majority

NPM : node package manager

Link: Tài Liệu

PHẦN 1: Domain name & Hosting - Git (option) - Docker (option)

Domain name & Hosting

Lab 1. Hiểu về DNS

Domain Name System : hệ thống chứa tên miền, cung cấp dịch vụ chuyển đổi tên miền - ip và ip - tên miền. ***IP ở đây chỉ liên quan như là thay thế tên miền. Còn có địa chỉ IP DNS riêng. ví dụ địa chỉ ip DNS google là 8.8.8.8***

Tên miền cấp 1 (quốc tế) : .org, .com

Tên miền cấp 2 (quốc gia) : .vn, .cn

Tên miền cấp 3 (quốc tế-gia) : .com.vn

**** Một địa chỉ ip có thể gán nhiều tên miền****

Lab 2. Đăng ký tên miền miễn phí

Lab 3. Đăng ký tên miền có phí

Lab 4. Hiểu về web server

Web server là gồm phần cứng, hệ điều hành, phần mềm web server chứa nội dung web.

**** Vị trí tập tin được dẫn bằng địa chỉ URL trên thanh địa chỉ trình duyệt (Uniform Resource Locator) gồm các phần**

- Scheme (http, https, ftp): giao thức giao tiếp web client -web server. Nếu không trình duyệt mạng thì sẽ ko có giao thức scheme mà chỉ có “ file đầu địa chỉ trình duyệt trên máy tính cá nhân.
- Tên miền:
- Cổng:
- Path: đường dẫn tới thư mục sau khi tới được tên miền

Ví dụ:

https://quantrimang.com/dia-chi-ip-va-dia-chi-mac-hoat-dong-song-song-nh
u-the-nao-158460 ******

Lab 5. Cài đặt và cấu hình XAMPP

Xampp là 1 chương trình tạo web server có chứa sẵn phần mềm web server, nhưng máy tính khác không thể truy cập web server này vì đây chỉ là web server cục bộ (chưa đăng ký tên miền bla bla).

- X (hệ điều hành), A (Apache: xử lý yêu cầu gửi tới máy chủ qua giao thức http), M (Tên cũ của Mysql), P (php), P(biên dịch mã nguồn Perl).
- Phần mềm web server như là: Apache, IIS.

****Các phần mềm tương tự như Xampp là: Wamp, Lamp.****

Lab 6. Tìm hiểu về các loại hosting

Hosting là một thư mục đặt nội dung trang web vào. Cần liên kết tên miền đã đăng ký vào hosting này để truy cập

Các loại hosting: share hosting, cloud hosting...

Lab 7. Tìm hiểu các thông số liên quan đến một web hosting

Bảng thông, dung lượng lưu trữ, bảo mật....

Lab 8. Shared hosting miễn phí

Shared hosting là chia sẻ tài nguyên hosting của máy chủ server cho các cá nhân, tổ chức cần đặt nội dung website của mình lên web server.

Lab 9. Shared hosting có phí

Với nhu cầu **xem lại** tập tin hay **khôi phục** lại trạng thái ban đầu của tập tin => cần dùng tới hệ thống quản lý phiên bản (version) cho các tập tin / dự án....
Được gọi là Version Control System.

Chức năng chính của hệ thống quản lý phiên bản:

- Lưu trữ được các trạng thái, nội dung thay đổi của các tập tin
- Biết được ai là người chỉnh sửa/ nộp tập tin
- Biết được tập tin đã được thay đổi những gì
- Có thể khôi phục lại các tập tin đã bị xóa

Version Control System được chia làm 3 loại:

- **Hệ thống cục bộ:** taptin1 - taptin2 - taptin3, viet_1/1/2020 - viet 2/1/2020.
- **Hệ thống tập trung:** taptin1 - taptin2 - taptin3 và danh sách các máy client có thể lấy các tập tin về, sau đó chỉnh sửa và cập nhật lại cho máy chủ. => ?? ***có biết được ai sửa không, có lưu được trạng thái thay đổi không? còn khôi phục lại bản cũ chắc là ko dc! => vậy các chức năng chính của hệ thống quản lý phiên bản trên là của 3 loại hay chỉ phân tán ???***

- **Hệ thống phân tán:** các máy client có thể lấy toàn bộ cả kho chứa các tập tin, lấy được phiên bản mới nhất

Git (option) : hệ thống quản lý phiên bản phân tán

Version Control systems

Lab 10. Tải và cài đặt Git

Cấu hình môi trường làm việc: **danh tính, chương trình soạn thảo văn bản**
 Với git, bạn có thể lưu thông tin ở 3 nơi khác nhau tương ứng với 3 mức cấu hình:

- system: thông tin và kho chứa được dùng cho tất cả mọi người: Ví dụ: `$ git config --system user.name "Van Teo"`
- global: thông tin và kho chứa sẽ được dùng cho người đang đăng nhập: Ví dụ: `$ git config --global user.name "Van Teo"`
- local: thông tin chỉ được dùng duy nhất cho 1 kho chứa: Ví dụ: `$ git config user.name "Van Teo"`

Lab 11. Cấu hình Git lần đầu

Tạo danh tính: tên & email => dùng để biết ai commit

Mục đích	Lệnh
Cấu hình tên và email	<code>\$ git config --global user.name "Phan An"</code> <code>\$ git config --global user.email Nongphanan762@gmail.com</code>
Cấu hình chương trình soạn thảo cho git (visual studio code)	<code>\$ git config --global core.editor code</code>
Kiểm tra thông tin cấu hình	<code>\$ git config --list</code>

Lab 12. Hiểu về Working Directory, Staging Area và Git Directory

Có 2 loại kho chứa:

- kho chứa cục bộ:

Khi tạo kho chứa (nơi hệ thống quản lý phiên bản làm việc) sẽ tạo 3 phần: **working directory, staging area, git directory.**

Lab 13. Tạo một thư mục có tên project, tạo nội dung và thực hiện các thao tác cần thiết để ra được kết quả.

Lab 14. Thực hành với tập tin .gitignore

Lab 15. Hiểu về các tình huống xóa tập tin

Lab 16. Các chế độ xem lịch sử commit.

Lab 17. Lọc các commit theo yêu cầu.

Lab 18. Các thao tác hủy bỏ.

Lab 19. Tạo kho chứa trên Github

Lab 20. Tạo kho chứa trên Gitlab

Lab 21. Thực hiện đẩy local repo lên remote repo

Lab 22. Thực hiện đẩy local repo lên remote repo (tt)

Lab 23. Thêm kho chứa ở xa

Lab 24. Thực hiện một số thao tác liên quan đến kho chứa ở xa.

Lab 25. Thao tác với Tag.

Lab 26. Làm việc với nhánh

Lab 27. Hiểu về phân nhánh và fast-forward.

Lab 28. Tạo ra tình huống có xung đột khi tích hợp