

Web server: Nginx + Apache + OctoberCMS

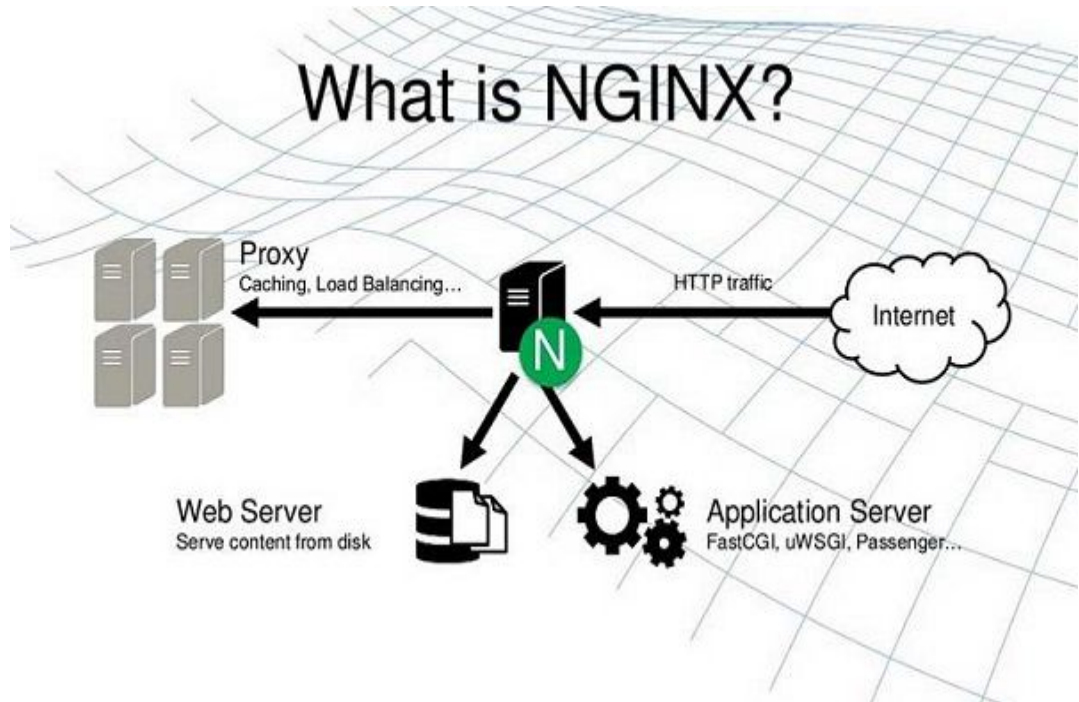
I. Danh sách thành viên & Công việc

Họ & tên	MSSV	Công việc	Tiến độ
Trần Anh Quyền	1651060674	<ul style="list-style-type: none">- Tìm hiểu tài liệu về cách cài đặt web server: NGINX + Apache + OctoberCMS trên máy ảo cục bộ UbuntuServer- Cài đặt demo website sử dụng mã nguồn mở OctoberCMS bao gồm:<ul style="list-style-type: none">+, Cài đặt nginx (port 80)+, Cài đặt apache (port 8080)+, Cài đặt mã nguồn mở OctoberCMS.	<ul style="list-style-type: none">- Đã hoàn thành
Lê Viết Quyết	1651061178	<ul style="list-style-type: none">- Cài đặt UbuntuServer- Tìm hiểu tài liệu về:<ul style="list-style-type: none">+, Nginx+, Apache+, OctoberCMS- Cài đặt Nginx (port 80)- Cài đặt Apache (port 8080)+, Cài đặt mã nguồn mở OctoberCMS.	<ul style="list-style-type: none">- Đã hoàn thành

II. Nội dung nghiên cứu

1. Nginx, Apache và OctoberCMS là gì?

a. Nginx là gì?



- *NGINX là một phần mềm web server mã nguồn mở, sử dụng kiến trúc hướng sự kiện (event-driven) không đồng bộ (asynchronous).* Mục tiêu ban đầu để phục vụ **HTTP cache** nhưng sau được áp dụng vào **reverse proxy**, **HTTP load balancer** và các giao thức truyền mail như **IMAP4**, **POP3**, và **SMTP**. **NGINX** chính thức ra đời vào tháng 10/2014. Đây là phần mềm giúp server có tốc độ và khả năng mở rộng lớn nhất, đồng thời, xử lý và thao tác trên hàng nghìn kết nối cùng lúc. Do đó, rất nhiều “ông lớn” công nghệ hiện nay đều lựa chọn **NGINX** như **Google**, **Adobe**, **Netflix**, **WordPress**...

b. **Apache** là gì?



- **Apache** là **phần mềm web server** miễn phí mã nguồn mở. Hiện nay nó đang chiếm đến khoảng 46% thị phần websites trên toàn thế giới. Với tên gọi chính thức là **Apache HTTP Server**, **Apache** được điều hành và phát triển bởi **Apache Software Foundation**. Mục đích của **Apache** là giúp chủ trang website đưa nội dung lên web. Do đó **Apache** được xếp vào dạng phần mềm “**web server**”. Đây là một trong số những **web server** lâu đời và đáng tin cậy nhất với phiên bản đầu tiên đã được ra mắt từ hơn 20 năm trước. **Apache** là một trong những web server phổ biến nhất thế giới cho phép bạn thiết lập website an toàn, tiết kiệm chi phí lẫn công sức một cách hiệu quả và được những đơn vị kinh doanh vừa và nhỏ ưu tiên lựa chọn hàng đầu. **Apache** đóng một vai trò quan trọng trong quá trình phát triển của **mạng web thế giới www**.

c. **OctoberCMS** là gì?

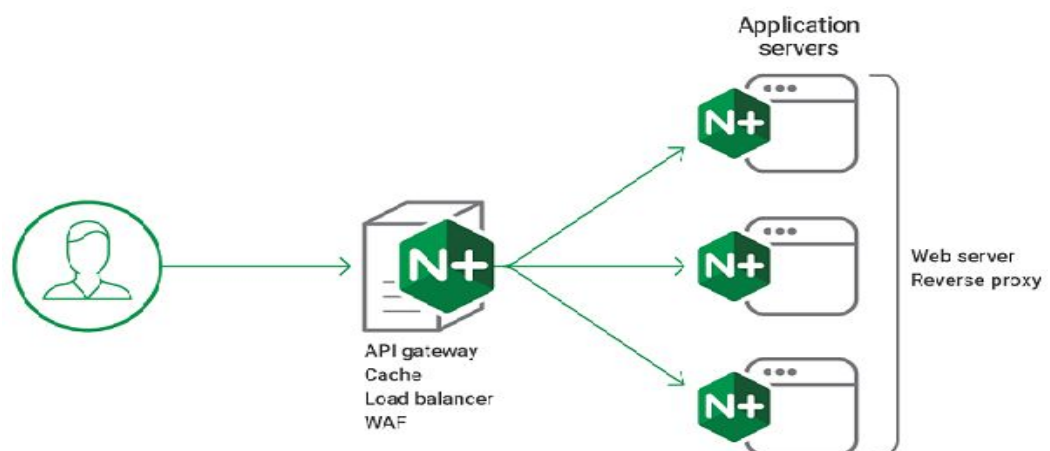
- **CMS** (Content Management System) - Hệ quản trị nội dung là một hệ thống phần mềm nhằm quản lý nội dung, chúng tạo ra môi trường để tổ chức hoặc cộng tác thuận lợi hơn cho việc xây dựng một hệ thống tài liệu, nội dung có cấu trúc.



- **OctoberCMS** là hệ quản trị nội dung mã nguồn mở được xây dựng trên nền tảng Laravel ra đời từ năm 2014. OctoberCMS kế thừa được toàn bộ sức mạnh của Laravel (framework PHP số 1 hiện nay) kết hợp với một thiết kế mở, giúp xây dựng những website chạy rất nhanh, mạnh mẽ và đặc biệt là xây dựng những tính năng riêng dễ dàng. Là một CMS đa năng mạnh mẽ với cộng đồng lập trình viên xây dựng mạnh mẽ October cho thấy một viễn tưởng sáng giá cho CMS thay thế WordPress trong thời gian tới. Dựa trên Framework Laravel do đó nó là CMS phổ biến đầu tiên được các lập trình viên quen thuộc Laravel chọn cho dự án sắp tới của mình.

2. Nginx, Apache và OctoberCMS có tính năng cụ thể là gì?

a. Các tính năng của Nginx



- Có thể xử lý hơn 10.000 kết nối cùng lúc với bộ nhớ thấp.
- Phục vụ tập tin tĩnh (static files) và lập chỉ mục tập tin.
- Tăng tốc proxy ngược bằng bộ nhớ đệm (cache); cân bằng tải đơn giản và khả năng chịu lỗi.
- Hỗ trợ tăng tốc với bộ nhớ đệm của FastCGI, uWSGI, SCGI, và các máy chủ memcached.
- Kiến trúc modular.
- Hỗ trợ mã hoá SSL và TLS.
- Cấu hình linh hoạt, lưu lại nhật ký truy vấn.
- Chuyển hướng lỗi 3XX-5XX.
- Rewrite URL (URL rewriting) dùng regular expressions.
- Hạn chế tỷ lệ đáp ứng truy vấn.
- Giới hạn số kết nối đồng thời hoặc truy vấn từ 1 địa chỉ.
- Khả năng nhúng mã PERL.
- Hỗ trợ và tương thích với IPv6.
- Hỗ trợ WebSockets.
- Hỗ trợ truyền tải file FLV và MP4.

b. Một số tính năng phổ biến của **Apache**



- .htaccess
- Tương thích IPv6
- FTP
- Tương thích HTTP / 2

- Perl, Lua và PHP
- Điều chỉnh băng thông
- WebDAV
- Load balancing
- URL rewriting
- Session tracking
- Định vị địa lý dựa trên địa chỉ IP
- Nhiều chế độ xử lý yêu cầu (MPM) bao gồm Event driven / Async , Threaded và Prefork.
- Xử lý các tệp tĩnh, tệp chỉ mục, tự động lập chỉ mục và nội dung
- TLS / SSL với hỗ trợ stapling SNI và OCSP , thông qua OpenSSL hoặc wolfSSL .
- Kiểm soát truy cập xác thực và ủy quyền chi tiết
- Nén và giải nén gzip
- Hạn chế kết nối đồng thời
- Điều chỉnh băng thông
- WebDAV
- Hỗ trợ CGI
- Trình phân tích cú pháp biểu thức chung
- Chế độ xem trạng thái thời gian thực
- Hỗ trợ XML
- Hỗ trợ FTP (bằng một mô-đun riêng)

c. Một số tính năng của **OctoberCMS**



- OctoberCMS là sự kết hợp giữa các tập tin (flat file) và cơ sở dữ liệu CSDL, mọi thứ được lưu nhiều hơn dưới dạng file tĩnh. Do vậy, các nội dung có thể được tạo ra rất nhanh và bạn có thể chỉnh sửa các thực thể này trong qua backend một cách dễ dàng. Nếu cần các nội dung động, bạn có thể lưu trữ trong các bảng dữ liệu riêng, nhờ đó bạn có toàn quyền tạo ra các chức năng riêng biệt. OctoberCMS hỗ trợ các hệ quản trị CSDL như MySQL, SQLite và PostgreSQL.
- October CMS được tích hợp sẵn tính năng phân quyền, chỉnh sửa đa thành phần nội dung thông qua các tab và các phím tắt.
- Khi sử dụng OctoberCMS cho một ứng dụng web phức tạp, bạn có toàn quyền tự do thực hiện mọi thứ trên đó. Phần backend tự động xây dựng cho bạn phần danh sách và form nhập liệu cho mỗi model bạn tạo ra, bạn có thể điều chỉnh hoàn toàn theo ý mình thông qua các tùy chỉnh. Bạn cũng có thể tạo ra các widget trong backend với hình thức riêng dễ dàng, nó làm cho backend có tính tùy biến cao, cải thiện khả năng sử dụng khi chỉnh sửa các nội dung phức tạp.

3. Tại sao phải cài đặt song song NGINX + Apache, Hoạt động của NGINX + Apache + OctoberCMS?

a. Tại sao phải cài đặt song song NGINX + Apache

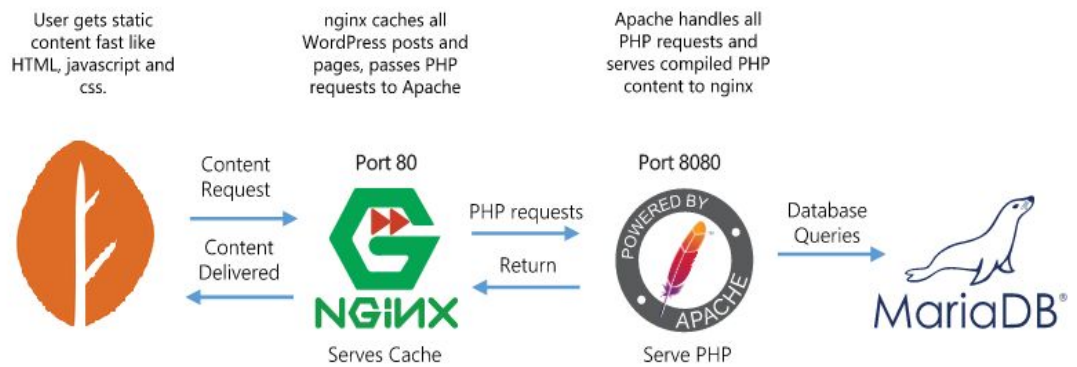


- Người ta thường sử dụng Apache bởi vì Apache tỏ ra tốt hơn Nginx trong việc phục vụ các trang web động (dynamic page). Nhưng vì tính đa dụng nên Apache có rất nhiều thành phần “thừa” khiến cho Web Server của chúng ta trở nên chậm chạp. Trong khi Nginx là một máy chủ web nhỏ nhẹ, hiệu năng cao, miễn phí.
- Nginx và Apache là sự kết hợp hoàn hảo và phổ biến giúp tăng tốc website, tăng khả năng chịu tải và tối ưu cho server nhờ tận dụng được ưu thế của cả 2 web server Apache (xử lý trang động: php) và Nginx (xử lý trang tĩnh: html, zip, image,...)
- Dùng Nginx như là một Proxy ngược cho Apache là một trong những cách thông thường nhất để sử dụng Nginx giúp tăng tốc website và tăng khả năng chịu tải của server. Nginx sẽ chạy ở phía trước (front-end) phục vụ port 80 xử lý các trang web tĩnh, còn Apache sẽ chạy ở phía sau (back-end) lắng nghe trên port 8080. Các Client truy vấn tới Web Server sẽ theo dạng:

Client <—> Nginx <—> Apache

b. Hoạt động của NGINX + Apache + OctoberCMS

Nginx Reverse Proxy + Apache + OctoberCMS



- Như chúng ta đã biết, trình duyệt sẽ đọc dữ liệu từ server truyền về thông qua cổng 80 và mặc định khi cài NGINX hay Apache nó cũng đều được sử dụng cổng này. Nhưng bây giờ, chúng ta sẽ cho Apache cho một cổng nào đó (8080 chẳng hạn) mà trình duyệt sẽ không đọc trực tiếp được, rồi chúng ta sẽ sử dụng cổng 80 cho NGINX, lúc này NGINX sẽ tự động gửi các truy vấn từ các file có đuôi mở rộng là .php đến cổng của Apache cho nó xử lý rồi Apache trả dữ liệu lại cho NGINX rồi NGINX gửi cho người dùng đọc.

III. Hướng dẫn cài đặt **NGINX + Apache + OctoberCMS** trên **UbuntuServer**

- Chuẩn bị: Server ảo Ubuntu 18.04
- Các phần mềm cần cài : **Apache 2, MariaDB Database Server, PHP 7.02 trở lên, Nginx, OctoberCMS.**

1. Tiến hành cài đặt

a. Cài đặt Apache2

+ Cú pháp:

```
sudo apt update
```

```
sudo apt-get install apache2
```

+ Sau khi cài xong dùng các lệnh sau để bắt đầu chạy, dừng, khởi động cùng server cho Apache:

```
sudo systemctl stop apache2.service
```

(Dừng chạy Apache)

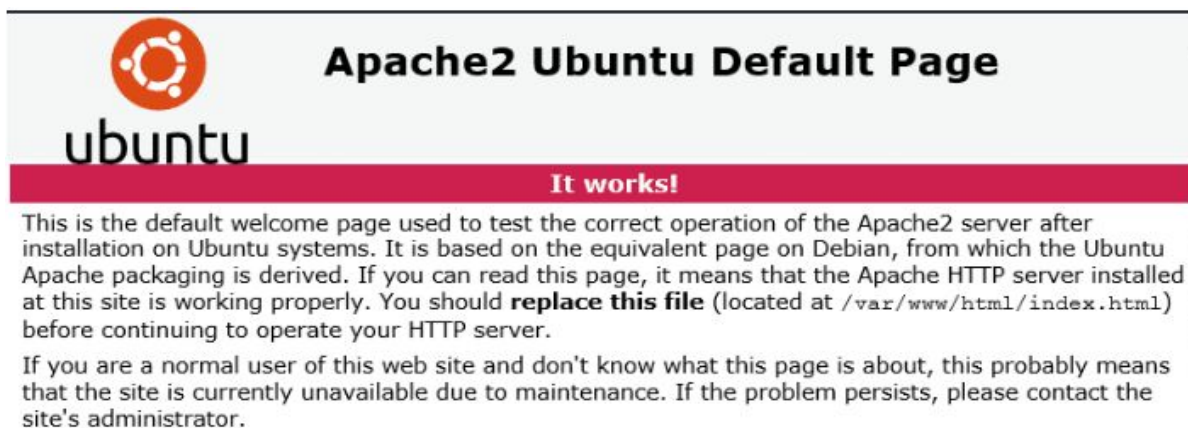
```
sudo systemctl start apache2.service
```

(Khởi chạy Apache)

```
sudo systemctl enable apache2.service
```

(Thiết lập chạy cùng server)

- + Để kiểm tra việc cài đặt Apache đã hoàn thành hay chưa, ta vào trình duyệt nhập server hostname hoặc địa chỉ IP của server, ta sẽ thấy trang web mặc định của Apache như hình dưới:



b. Cài đặt MariaDB

- + **Cú pháp:**

`sudo apt-get install mariadb-server mariadb-client`

- + Sau khi cài đặt xong ta sử dụng các lệnh bên dưới để chạy, dừng và thiết lập cho MariaDB khởi chạy cùng server:

`sudo systemctl stop mariadb.service`

(Dừng chạy MariaDB)

`sudo systemctl start mariadb.service`

(Bắt đầu chạy MariaDB)

`sudo systemctl enable mariadb.service`

(Cho phép MariaDB khởi động cùng server)

- + Sau đây, chạy các lệnh bên dưới để bảo mật MariaDB bằng cách tạo mật khẩu root, và không cho phép truy cập root từ xa:

`sudo mysql_secure_installation`

- + Khi được nhắc, hãy trả lời các câu hỏi dưới đây bằng cách làm theo hướng dẫn:
 - Enter current password for root (enter for none): Just press the **Enter**
 - Set root password? [Y/n]: **Y**
 - New password: **Enter password**
 - Re-enter new password: **Repeat password**
 - Remove anonymous users? [Y/n]: **Y**
 - Disallow root login remotely? [Y/n]: **Y**
 - Remove test database and access to it? [Y/n]: **Y**
 - Reload privilege tables now? [Y/n]: **Y**
- + Để kiểm tra MariaDB đã được cài đặt hay chưa, sử dụng cú pháp:

```
sudo mysql -u root -p
```

Sau đó nhập mật khẩu bạn đã tạo ở trên để đăng nhập trên nếu thành công, bạn sẽ thấy thông báo MySQL welcome

c. Cài đặt PHP 7.2 và các Modules liên quan

- + Chạy các lệnh bên dưới để thêm vào kho và nâng cấp lên PHP 7.2

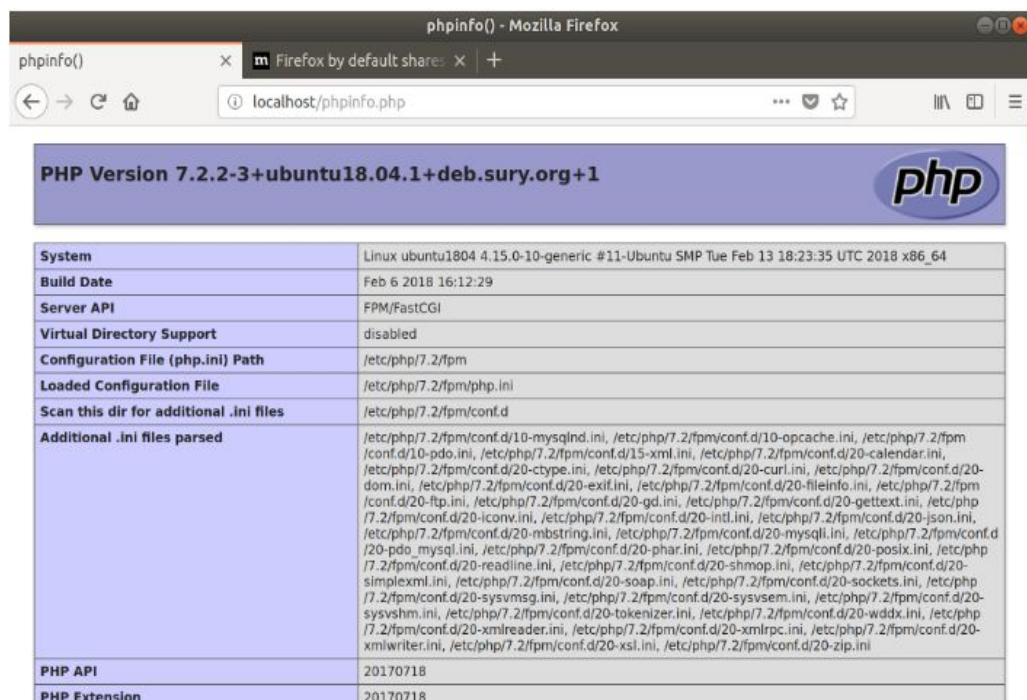

```
sudo apt-get install software-properties-common
sudo add-apt-repository ppa:ondrej/php
```
- + Tiếp theo chạy các lệnh bên dưới để cài đặt PHP 7.2 và các modules:


```
sudo apt install php7.2 libapache2-mod-php7.2
php7.2-common php7.2-curl php7.2-intl php7.2-mbstring
php7.2-xmlrpc php7.2-mysql php7.2-gd php7.2-pgsql
php7.2-xml php7.2-cli php7.2-imagick php7.2-bcmath
php7.2-gmp php7.2-zip
```
- + Sau khi cài đặt xong PHP 7.2, chạy lệnh bên dưới để file cấu hình mặc định PHP cho Apache2:


```
sudo nano /etc/php/7.2/apache2/php.ini
```
- + Sau đó thay đổi trong file các nội dung tương ứng như sau :


```
file_uploads = On
allow_url_fopen = On
short_open_tag = On
memory_limit = 256M
upload_max_filesize = 100M
max_execution_time = 360
date.timezone = Asia/Ho_Chi_Minh
```

- + Lưu và đóng file lại. Rồi restart Apache2.
- + Để test PHP 7.2 cài đặt với Apache2, ta tạo 1 file phpinfo.php trong thư mục gốc của Apache2 với câu lệnh sau:
`sudo nano /var/www/html/phpinfo.php`
- + Nhập nội dung dưới đây vào:
`<?php phpinfo(); ?>`
- + Lưu file sau đó chạy trên trình duyệt với tên miền /phpinfo.php
- + Bạn sẽ thấy trang web mặc định



PHP Version 7.2.2-3+ubuntu18.04.1+deb.sury.org+1	
System	Linux ubuntu1804 4.15.0-10-generic #11-Ubuntu SMP Tue Feb 13 18:23:35 UTC 2018 x86_64
Build Date	Feb 6 2018 16:12:29
Server API	FPM/FastCGI
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/7.2/fpm
Loaded Configuration File	/etc/php/7.2/fpm/php.ini
Scan this dir for additional .ini files	/etc/php/7.2/fpm/conf.d
Additional .ini files parsed	/etc/php/7.2/fpm/conf.d/10-mysqlnd.ini, /etc/php/7.2/fpm/conf.d/10-opcache.ini, /etc/php/7.2/fpm/conf.d/10-pdo.ini, /etc/php/7.2/fpm/conf.d/15-xml.ini, /etc/php/7.2/fpm/conf.d/20-calendar.ini, /etc/php/7.2/fpm/conf.d/20-curl.ini, /etc/php/7.2/fpm/conf.d/20-dom.ini, /etc/php/7.2/fpm/conf.d/20-exif.ini, /etc/php/7.2/fpm/conf.d/20-fileinfo.ini, /etc/php/7.2/fpm/conf.d/20-ftp.ini, /etc/php/7.2/fpm/conf.d/20-gd.ini, /etc/php/7.2/fpm/conf.d/20-gettext.ini, /etc/php/7.2/fpm/conf.d/20-iconv.ini, /etc/php/7.2/fpm/conf.d/20-intl.ini, /etc/php/7.2/fpm/conf.d/20-json.ini, /etc/php/7.2/fpm/conf.d/20-mbstring.ini, /etc/php/7.2/fpm/conf.d/20-mysqli.ini, /etc/php/7.2/fpm/conf.d/20-pdo_mysql.ini, /etc/php/7.2/fpm/conf.d/20-phar.ini, /etc/php/7.2/fpm/conf.d/20-posix.ini, /etc/php/7.2/fpm/conf.d/20-readline.ini, /etc/php/7.2/fpm/conf.d/20-shmop.ini, /etc/php/7.2/fpm/conf.d/20-simplexml.ini, /etc/php/7.2/fpm/conf.d/20-soap.ini, /etc/php/7.2/fpm/conf.d/20-sockets.ini, /etc/php/7.2/fpm/conf.d/20-sysvmsg.ini, /etc/php/7.2/fpm/conf.d/20-sysvsem.ini, /etc/php/7.2/fpm/conf.d/20-sysvshm.ini, /etc/php/7.2/fpm/conf.d/20-tokenizer.ini, /etc/php/7.2/fpm/conf.d/20-wddx.ini, /etc/php/7.2/fpm/conf.d/20-xmlreader.ini, /etc/php/7.2/fpm/conf.d/20-xmlrpc.ini, /etc/php/7.2/fpm/conf.d/20-xmlwriter.ini, /etc/php/7.2/fpm/conf.d/20-xsl.ini, /etc/php/7.2/fpm/conf.d/20-zip.ini
PHP API	20170718
PHP Extension	20170718

d. Cài đặt Nginx

- + Cú pháp:

`sudo apt install nginx`

- + Config port cho Apache:

* Sửa lại port cho apache sang 8080

```
root@quyenquyet:~# nano /etc/apache2/ports.conf
```

```
# If you just change the port or add more ports here, you will likely also
# have to change the VirtualHost statement in
# /etc/apache2/sites-enabled/000-default.conf

Listen 8080

<IfModule ssl_module>
    Listen 8443
</IfModule>

<IfModule mod_gnutls.c>
    Listen 8443
</IfModule>

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
```

Get Help Write Out Where Is Cut Text Justify Cur Pos Undo
Exit Read File Replace Uncut Text To Spell Go To Line Redo

* Restart apache

```
root@quyenquyet:~# systemctl restart apache2_
```

* Ktra trạng thái của apache

```
root@quyenquyet:~# systemctl status_apache2
```

* Kiểm tra port 8080 có đang chạy Apache không

```
root@quyenquyet:~# curl -I localhost:8080
```

```
root@quyenquyet:~# curl -I localhost:8080
HTTP/1.1 200 OK
Date: Sun, 19 Apr 2020 08:19:22 GMT
Server: Apache/2.4.29 (Ubuntu)
Last-Modified: Sun, 19 Apr 2020 08:11:43 GMT
ETag: "2aa6-5a3a0553a7c0c"
Accept-Ranges: bytes
Content-Length: 10918
Vary: Accept-Encoding
Content-Type: text/html

root@quyenquyet:~# _
```

+ Config lại nginx

```
root@quyenquyet:~# nano /etc/nginx/sites-available/default
```

```
# Read up on ssl_ciphers to ensure a secure configuration.
# See: https://bugs.debian.org/765782
#
# Self signed certs generated by the ssl-cert package
# Don't use them in a production server!
#
# include snippets/snakeoil.conf;

root /var/www/html;

# Add index.php to the list if you are using PHP
index index.html index.htm index.nginx-debian.html;

server_name _;

location / {
    proxy_pass http://127.0.0.1:8080;
    proxy_set_header Host $host;
    proxy_set_header X-Real-IP $remote_addr;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_set_header X-Forwarded-Proto $scheme;
}

# pass PHP scripts to FastCGI server
#
#location ~ \.php$ {
#    include snippets/fastcgi-php.conf;
#
#    # With php-fpm (or other unix sockets):
#    fastcgi_pass unix:/var/run/php/php7.0-fpm.sock;
#    # With php-cgi (or other tcp sockets):
#    fastcgi_pass 127.0.0.1:9000;
#}

# G Get Help  W Write Out  W Where Is  C Cut Text  J Justify  C Cur Pos  M-U Undo
# X Exit      R Read File  R Replace  U UnCut Text  T To Spell  G Go To Line M-F Redo
```


- + Restart nginx

```
root@quyenquyet:~# systemctl restart nginx_
```

- + Kiểm tra port 80

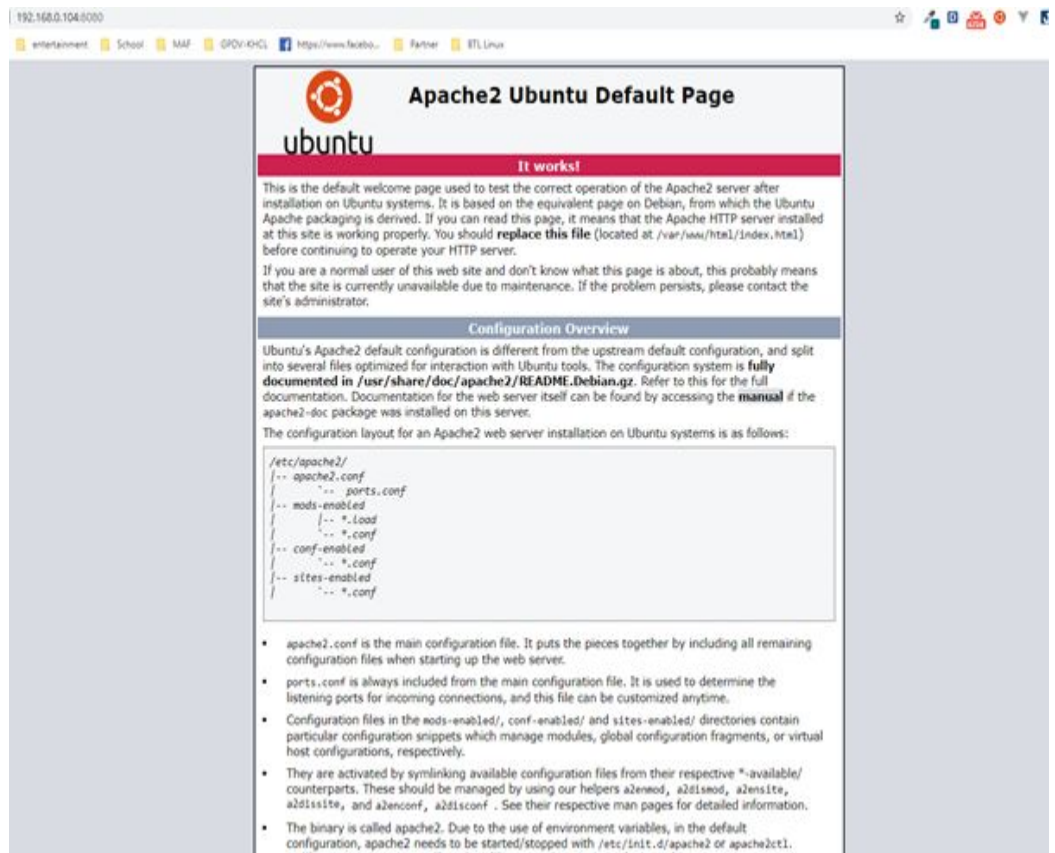
```
root@quyenquyet:~# curl -I localhost
HTTP/1.1 200 OK
Server: nginx/1.14.0 (Ubuntu)
Date: Sun, 19 Apr 2020 08:26:42 GMT
Content-Type: text/html
Content-Length: 10918
Connection: keep-alive
Last-Modified: Sun, 19 Apr 2020 08:11:43 GMT
ETag: "2aa6-5a3a0553a7c0c"
Accept-Ranges: bytes
Vary: Accept-Encoding

root@quyenquyet:~# _
```

```
root@quyenquyet:~# ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.0.104 netmask 255.255.255.0 broadcast 192.168.0.255
    inet6 fe80::a00:27ff:fe73:85bc prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:73:85:bc txqueuelen 1000 (Ethernet)
    RX packets 7 bytes 1136 (1.1 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 15 bytes 1726 (1.7 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 84 bytes 6324 (6.3 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 84 bytes 6324 (6.3 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

- + Truy cập bằng trình duyệt vào hostname hoặc địa chỉ ip máy chủ với port 8080.



e. Cài đặt OctoberCMS

- + Tạo Database cho OctoberCMS

```
sudo mysql -u root -p  
CREATE DATABASE october;
```

- + Tạo user octoberuser với password là “userpassword”

```
CREATE USER 'octoberuser'@'localhost'  
IDENTIFIED BY 'userpassword';
```

- + Cập quyền đầy đủ cho user này vào database

```
GRANT ALL ON october.* TO  
'octoberuser'@'localhost' IDENTIFIED BY  
'userpassword' WITH GRANT OPTION;
```

- + Lưu thay đổi và thoát

```
FLUSH PRIVILEGES;  
EXIT;
```

- + Tải OctoberCMS

```
cd /tmp && wget http://octobercms.com/download  
-O octobercms.zip
```

- + Sau khi tải, tạo 1 thư mục root cho OctoberCMS và giải nén tệp đã tải xuống vào thư mục gốc Apache2

- ```
sudo unzip octobercms.zip
sudo mv install-master /var/www/html/octobercms
```
- + Đặt quyền chính xác cho OctoberCMS hoạt động.
- ```
sudo chown -R www-data:www-data
/var/www/html/octobercms/
sudo chmod -R 755 /var/www/html/octobercms/
```
- + Config Virtual Host Apache
- ```
sudo nano /etc/apache2/sites-available
/octobercms.conf
```
- + Copy và paste nội dung sau vào file octobercms.conf
- ```
<VirtualHost *:80>
    ServerAdmin admin@example.com
    DocumentRoot /var/www/html/octobercms
    ServerName example.com
    ServerAlias www.example.com

    <Directory /var/www/html/octobercms/>
        Options +FollowSymLinks
        AllowOverride All
        Require all granted
    </Directory>

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log
    combined

</VirtualHost>
```
- + Sau khi định cấu hình Virtualhost ở trên, bật nó bằng cách chạy các lệnh bên dưới
- ```
sudo a2ensite octobercms.conf
sudo a2enmod rewrite
```
- + Restart Apache2
- ```
sudo systemctl restart apache2.service
```

+ Thử truy cập <http://example.com/install.php> trên máy thật.
(VD: máy windows)

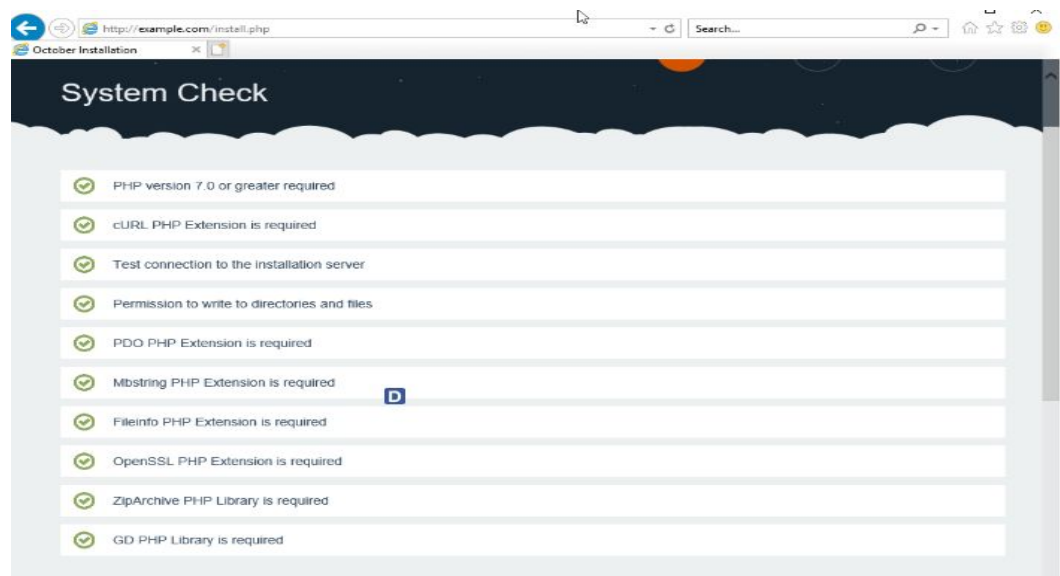
* Trước khi truy cập vào <http://example.com/install.php>, phải cài đặt host trên máy thật bằng cách:

Sửa file C:\Windows\System32\drivers\etc\hosts

Thêm dòng sau vào file

192.168.0.104 example.com

+ Truy cập vào <http://example.com/install.php>.



+ Cuối cùng thực hiện cài đặt database và tài khoản admin cho hệ thống

October Installation

GENERAL

Database

Administrator

ADVANCED

Advanced

Database

Please prepare an empty database for this installation.

Database Type: ☐

Please specify the database driver type for this connection.

MySQL Host:

Specify the hostname for the database connection.

MySQL Port:

(Optional) Specify a non-default port for the database connection.

Database Name:

Specify the name of the empty database.

MySQL Login:

User with all privileges in the database.

MySQL Password:

Password for the specified user.

Administrator >

October Installation

GENERAL

Database

Administrator

ADVANCED

Advanced

Administrator

Please specify details for logging in to the Administration Area.

First Name:

Last Name:

Email Address:

Admin Login:

Admin Password:

Confirm Password:

< Database Advanced >

Continue

October Installation

First blog post - October CMS

Dashboard | OctoberCMS

Dashboard CMS Media Blog Settings

WELCOME

Welcome back to OctoberCMS, Admin. Your last sign in was **Mon, Jan 1, 2018 11:16 PM**

[View access logs](#)

SYSTEM STATUS

Software is up to date

No warnings to display

System build 431

Event log 0

Request log 0

Online since January 1, 2018

Manage widgets

WEBSITE

October CMS

First blog post

Online Manage themes Customize theme

IV. Hướng dẫn Sử dụng/Quản trị.

Sau khi hoàn tất hết việc cài đặt các môi trường ở trên, điều ta quan tâm là làm sao để sử dụng và quản trị một hệ thống kết hợp như thế. Cụ thể hơn là làm sao để quản trị một trang tạo bởi **OctoberCMS**, có sử dụng kết hợp từ 2 server là **Nginx** và **Apache**. Chúng ta cùng xem hướng dẫn bên dưới để hiểu rõ hơn.

1. Làm quen với **OctoberCMS**

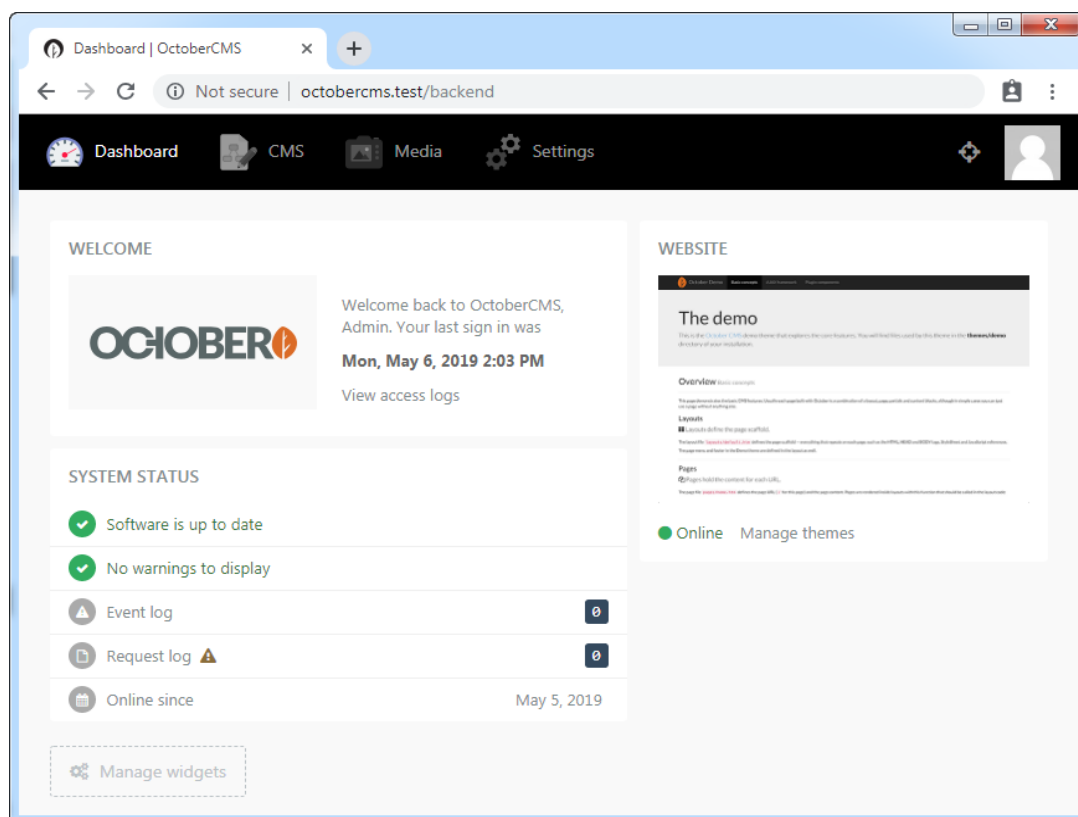
OctoberCMS cũng giống các CMS khác đều chia thành hai phần:

- **Frontend**: Chính là phần website của chúng ta, nơi các thông tin, hình ảnh được đưa lên để mọi người có thể đọc và tương tác với chúng.
- **Backend**: là vùng quản trị, nơi chúng ta quản lý nội dung (thêm, sửa, xóa).

Phần frontend phụ thuộc vào bạn chọn giao diện (theme) nào mà có cấu trúc và cách hiển thị khác nhau. Chúng ta tạm quên đi phần này mà cùng tìm hiểu về backend.

a. **Backend** - Khu vực quản trị CMS trong **OctoberCMS**

Mặc định để truy cập vào vùng quản trị ta truy cập vào đường dẫn yourdomain.com/backend. Lúc này sẽ xuất hiện trang quản trị như hình dưới đây:



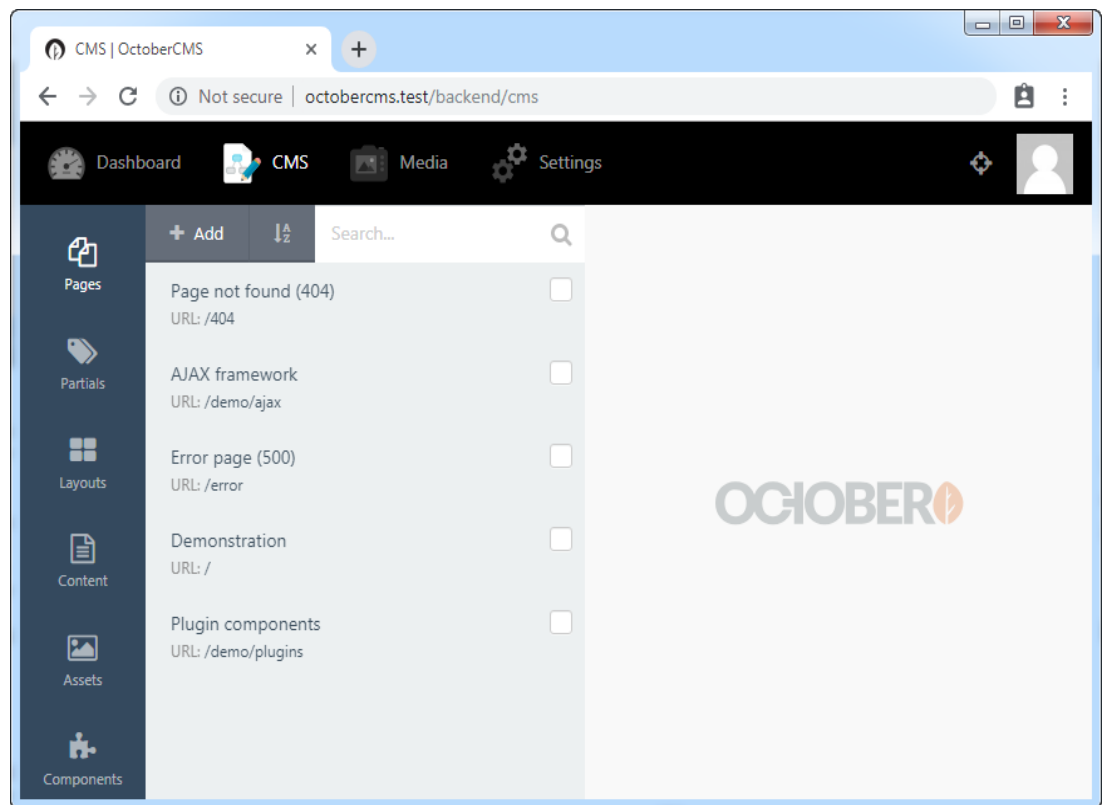
Như bạn đã thấy, giao diện bảng điều khiển (dashboard) của **OctoberCMS** khá đơn giản với trên cùng là thanh menu điều hướng bao gồm:

- **Dashboard:** Nơi hiển thị các thông tin điều khiển như thông tin tình trạng hệ thống (có bản cập nhật mới không, có cảnh báo lỗi nào không, thông tin về đăng nhập của tài khoản và giao diện hiện đang sử dụng). Các thành phần này gọi là các Widget. Chúng ta hoàn toàn có thể tạo ra các Widget này, ví dụ như tạo ra một thành phần hiển thị số lượng người xem các bài viết, số lượng đơn hàng hiện đang chưa xử lý...
- **CMS:** Đây chính là phần lõi tạo ra các trang web mà sẽ hiển thị ở frontend. Phần này sẽ là nơi chúng ta xử lý mọi vấn đề xoay quanh website.
- **Settings:** Tại đây chúng ta thiết lập các thông số, thuộc tính cho các thành phần của OctoberCMS.
- **Account (Ngoài cùng bên phải):** Nơi quản lý thông tin tài khoản.

Chúng ta sẽ cùng tìm hiểu cụ thể các thành phần trên trong phần tiếp theo. Dashboard khá đơn giản nên chúng ta sẽ bỏ qua vì ở đây chủ yếu hiển thị thông tin trạng thái hệ thống. Thành phần này sẽ được tìm hiểu sâu hơn trong phần thiết kế các Widget và tích hợp vào bảng điều khiển dashboard.

b. CMS thành phần cốt lõi của OctoberCMS backend

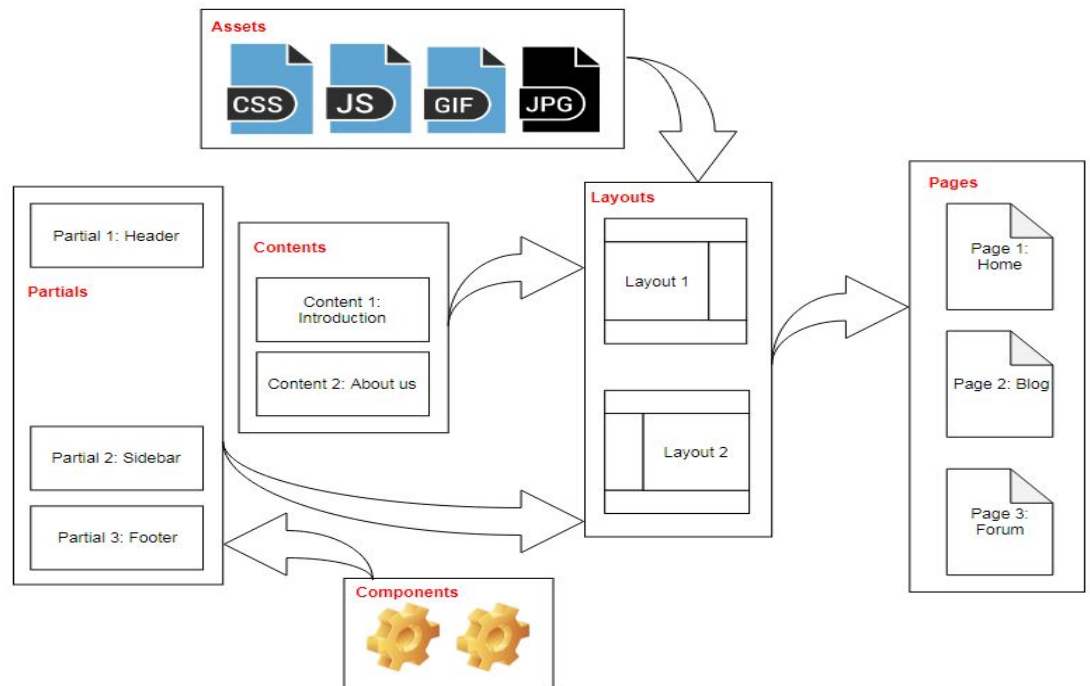
Đây là phần quan trọng nhất của OctoberCMS và nó cũng làm nên sự khác biệt với các hệ thống quản trị nội dung phổ biến hiện nay như Wordpress, Joomla, Drupal...



Giao diện CMS nơi làm việc thường xuyên nhất trong OctoberCMS được chia thành 3 phần:

- Thanh menu của CMS bao gồm các thành phần: Pages, Partials, Layouts, Content, Assets và Components.
- Phần diện tích tiếp theo là danh sách các phần tử con của từng mục trên menu CMS.
- Phần diện tích lớn nhất bên cạnh đó là nơi chúng ta sẽ thiết lập các thuộc tính, viết code...

Đây là phần hay nhất của OctoberCMS vì cách thực hiện mô đun hóa này giúp cho tạo ra các trang web nhanh chóng do sử dụng lại code hiệu quả. Chúng ta cùng tìm hiểu mô hình này:



Tại sao việc mô đun hóa như thế này hiệu quả, chúng ta cùng phân tích nhé. Đầu tiên trong website chúng ta sẽ có rất nhiều các trang ví dụ: trang chủ, trang bài viết, trang diễn đàn... Các trang này về nội dung là khác nhau nhưng về cấu trúc bố trí (**layout**) là giống nhau, ví dụ đầu tiên có phần **header** chứa thanh menu điều hướng, có phần chứa nội dung và phần sidebar chứa các thông tin phụ trợ, phần **footer** chứa các thông tin cuối trang như thông tin liên hệ.

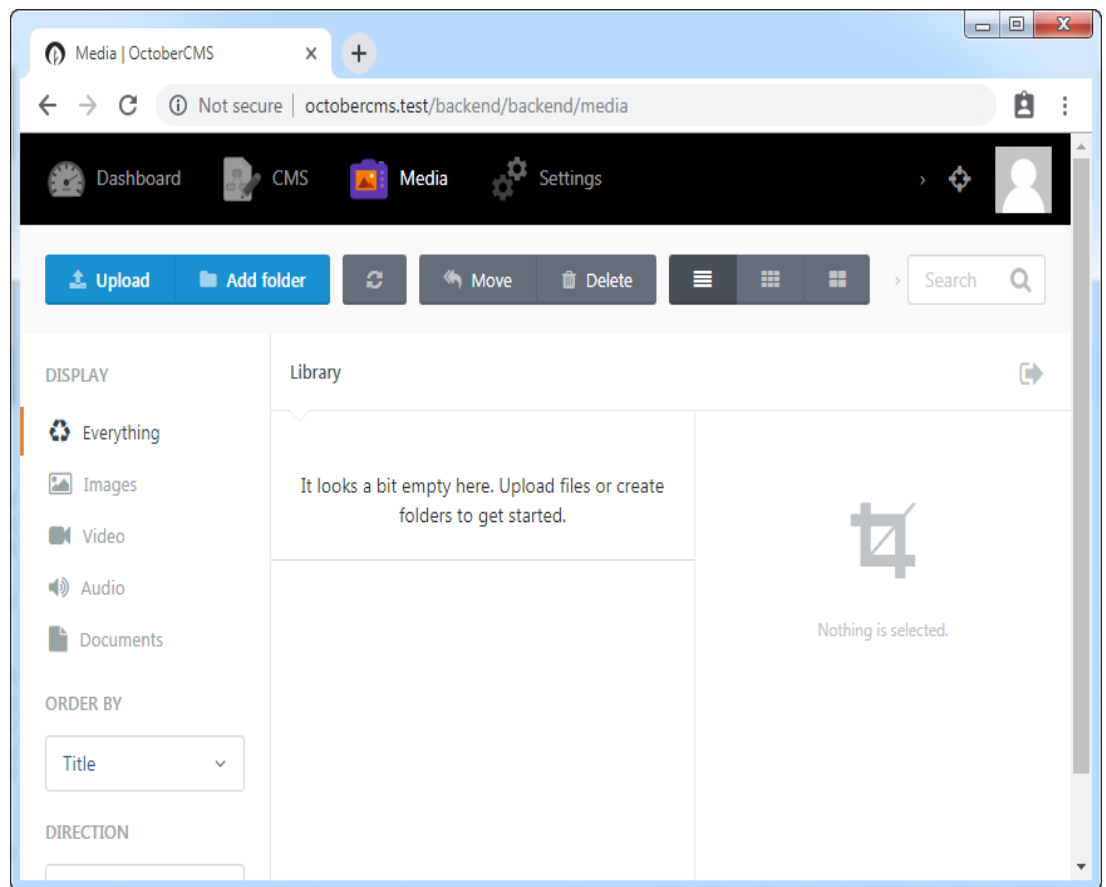
Trong hệ thống chúng ta có thể tạo ra nhiều **layout** khác nhau, với mỗi page khi áp dụng một **layout** cụ thể sẽ có thể hiện khác nhau.

Các **layout** lại được mô đun hóa thành các **partials**, **contents** với việc sử dụng các tài nguyên trong **assets** như hình ảnh, âm thanh, các file **css**, **javascript**.

Components là các thành phần chứa code có thể tạo ra các nội dung theo điều kiện logic, nó tác động và cung cấp dữ liệu cho **partials**.

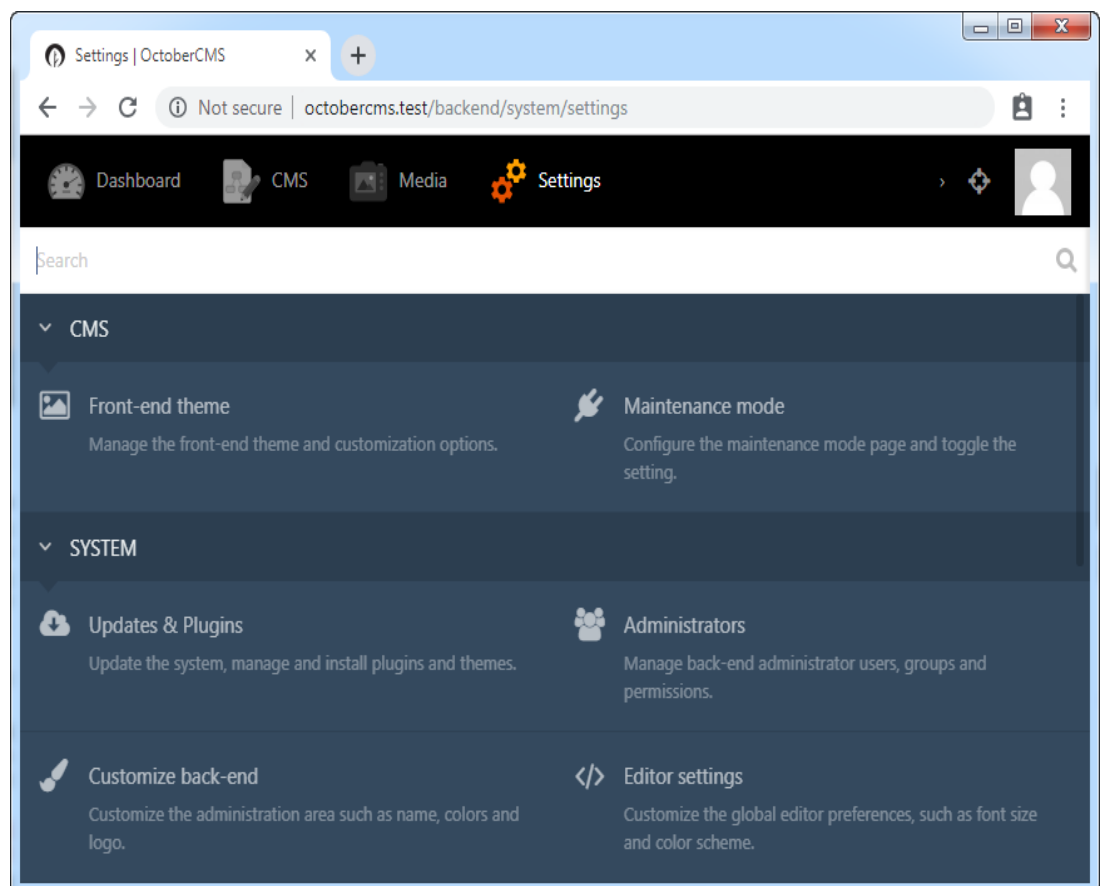
c. **Media** công cụ quản lý tài nguyên cho **OctoberCMS**

Cũng giống như các **CMS** khác, **OctoberCMS** cũng có phần quản trị các tài nguyên như ảnh, file âm thanh, video và các loại tài nguyên khác, cho phép tạo ra các cấu trúc thư mục để quản lý dễ dàng hơn.

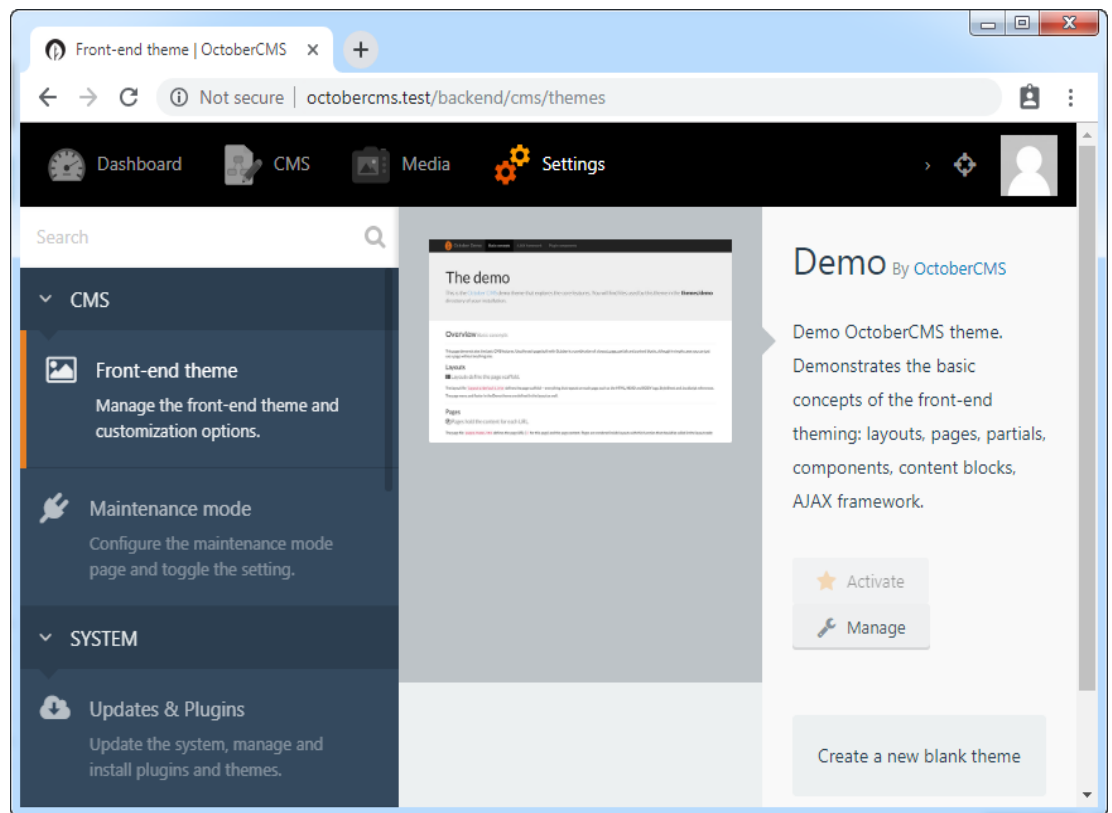


Với các tài nguyên như hình ảnh có sẵn công cụ đơn giản như crop ảnh.

d. Thiết lập, tùy chỉnh hệ thống trong **Settings**



Trong phần giao diện **settings** có hai phần hay dùng là thiết lập giao diện và quản lý cài đặt **plugin**.



Plugin là gói phần mềm cung cấp một chức năng nhất định cho OctoberCMS, ví dụ Rainlab.Blog là plugin cho phép các tính năng Blog cơ bản cho một website.

