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-nhu-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

Băng 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 dánh 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

Các thao tác:

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

Mục đích	Lệnh

Cấu hình tên và email	$ git config --global user.name "ngoctam123"
	
	$ git config --global user.email ngoctam.love1999@gmail.com

Cấu hình chương trình soạn thảo cho git (visual studio code)	$ git config --global core.editor code
Kiểm tra thông tin cấu hình	$ git config --list
Tạo kho chứa	$ git init

...or create a new repository on the command line

```
echo "# Web_MyPham" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/VoHang/Web_MyPham.git
git push -u origin main
```

Lab 12. Hiểu về thành phần kho chứa

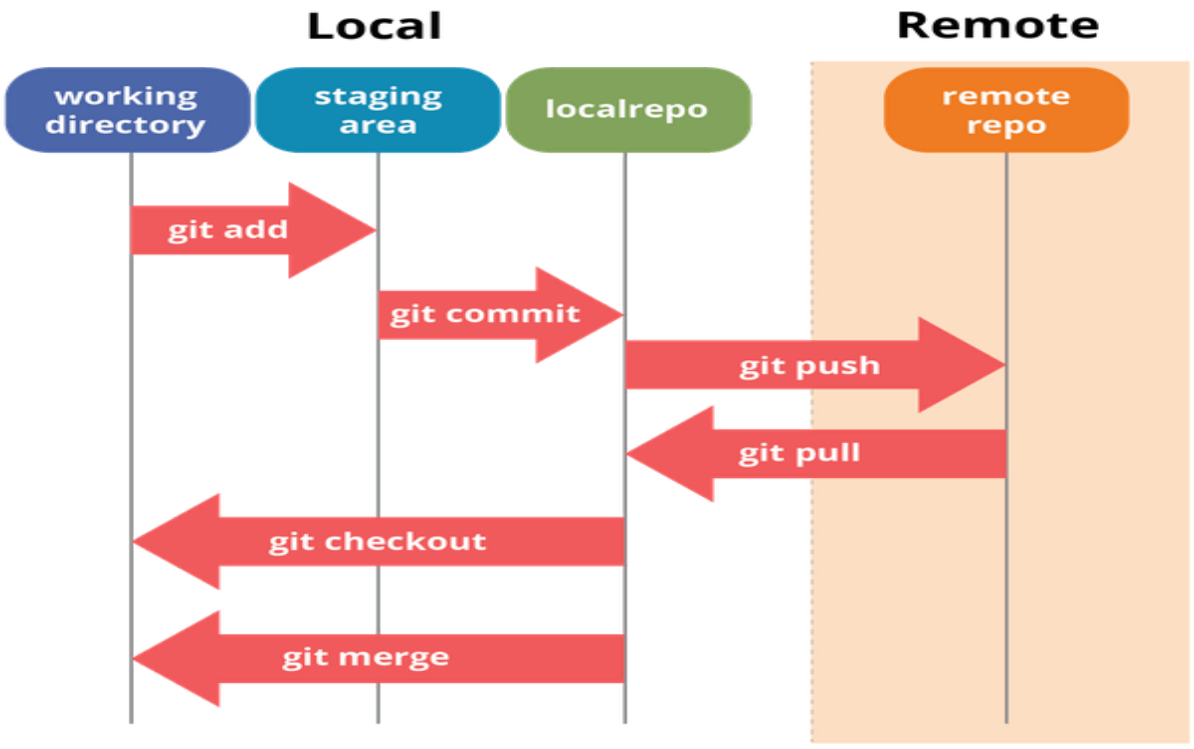
Thành phần

Có 2 loại kho chứa:

- Kho chứa cục bộ: kho chứa nằm trên máy tính cá nhân
- Kho chứa ở xa: nằm trên server như Github, Gitlab….

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

Working directory	**Khu vực làm việc, thư mục cần tạo kho chứa để quản lý, nằm cùng vs thư mục .git**
Staging area	**Các file đã được chỉnh sửa bỏ trong working directory**
Git directory (repository)	**Kho chứa**



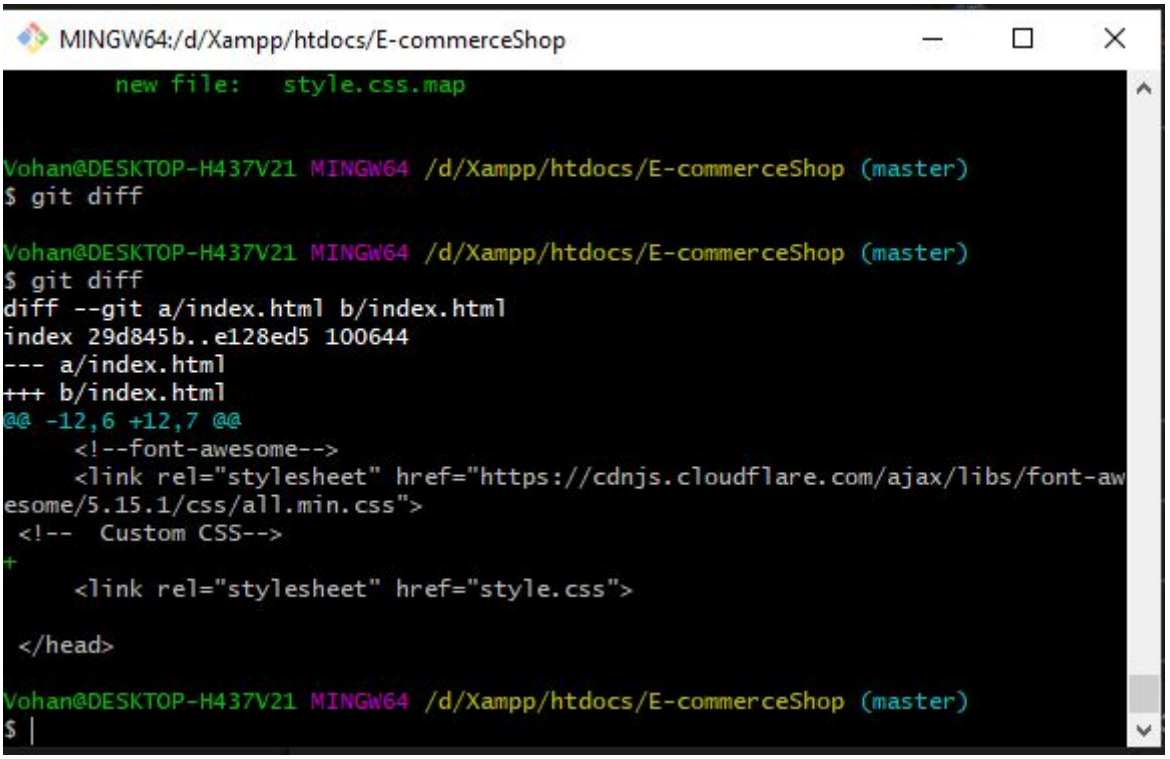
Sau khi git init -> các phần .git được xem là working directory
Các phần sau khi được chỉnh sửa trong working directory, hay git add -> staging area. ...như hình.

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.

Tạo kho chứa local

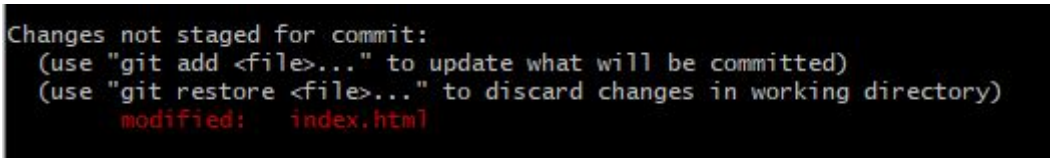
- Tạo remote repository ngoctam123/Web trên trang Github.com.
- Thực hiện lệnh git init cho thư mục web mỹ phẩm cần quản lý.
- Vào thư mục cần tạo kho chứa => ***Git bash here => git init***
- Cho phép kho chứa bắt đầu quản lý/ theo dõi tập tin..
- ***Git status***(xem trạng thái các tập tin, thư mục trong kho chứa. Git status chỉ có thể xem tập tin nào đã bị thay đổi mà không biết được đã thay đổi như thế nào)
- ***Git add tentaptin.***
- Nếu thêm ảnh vào thì sẽ không ảnh hưởng tới câu lệnh ***git diff***. Git diff sẽ thay đổi khi chỉnh sửa tập tin với điều kiện tập tin đó chưa được stage.

Nếu git diff có thay đổi sẽ:



(Minh họa: đã thay đổi trang index.html, phần thêm là dấu "+" màu xanh)

Khi đó git status sẽ báo index.html chưa dc xác định:



=>Modified index.html vì index.html đã bị đổi nhưng chưa được staged (add,....). Câu lệnh báo là Changes not staged for commit.Trong trường hợp tất cả tập tin đã được commit mà index.html chỉnh sửa thì sẽ báo

Changes to be committed

- Nếu có chỉnh sửa tập tin hay thư mục thì sau đó cần git add lại để theo dõi những thay đổi này.
- Đẩy lên kho chứa serve.

- Trước khi commit ta cần kiểm tra xem các tập tin nào đã được stage hay chưa để commit, dùng lệnh: ***Git diff --staged/cached***
- ***Git commit -m "Thông tin commit"***: Commit các tập tin để đưa vào localrepo chuẩn bị push lên kho chứa ở xa (github, gitlab).
- Thêm địa chỉ kho chứa ở xa cho local

  git remote add origin <URL của remote repo> // gắn địa chỉ của remote repo vào local repo

  git push -u origin master // đẩy lên nhánh master của remote repo

  git remote// xem danh sách kho chứa ở xa

  git remote rm tenkhochua // xóa kho chứa

Sao chép 1 kho chứa từ serve

Câu lệnh tạo: Git clone tên_kho_chứa.
Ví dụ: git clone https://github.com/ngoctam123/Web.git
Nếu muốn đổi tên kho chứa khi clone về có thể đặt tên mới sau câu lệnh:
Ví dụ: git init https://github.com/ngoctam123/Web.git tên_mới

Sao chép 1 kho chứa từ máy cục bộ

Việc sao chép này sẽ không tạo liên kết với kho chứa ở xa như clone.

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.

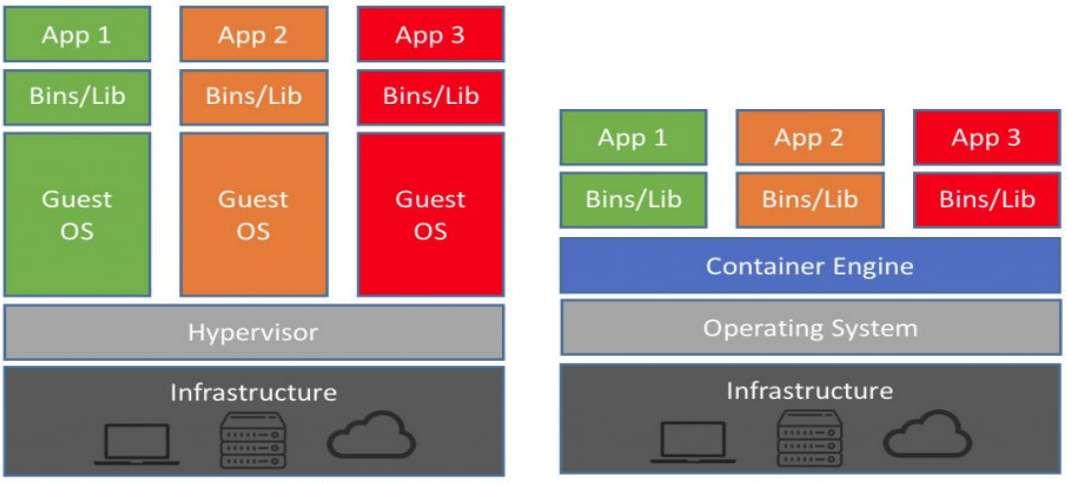
Lab 26. Làm việc với nhánh

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

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

Docker

Là 1 phần mềm cho phép người phát triển phần mềm dễ dàng chạy trên nhiều nền tảng khác nhau nhờ các container, các container giải quyết được vấn đề không đồng nhất môi trường giữa các server khác nhau.



=> Dễ quản lý, di chuyển

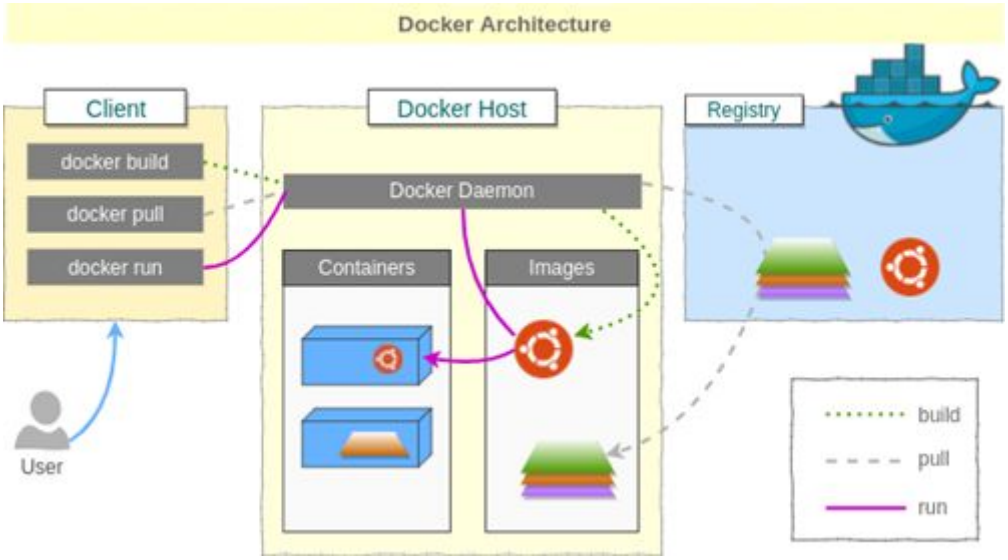
*Đầu tiên, sau khi cài docker, cần kích hoạt Hyper-V, vì docker có nhiều giải pháp để hoạt động, trong đó là dùng luôn hạ tầng Hyper. Mở PowerShell với quyền Administrator, gõ lệnh sau:*Enable-WindowsOptionalFeature -Online -FeatureName Microsoft-Hyper-V –All

So sánh máy ảo và docker

	Docker	Máy ảo
Giống		
Khác	- Chỉ có 1 hệ điều hành - Phần mềm chạy trên HĐH chủ - Dùng trực tiếp tài nguyên HĐH chủ -	- Chạy nhiều hệ điều hành - Phần mềm chạy trên HĐH ảo - HĐH ảo chiếm tài nguyên lớn - Hỗ trợ giao diện

Các thành phần của Docker

Một hệ thống Docker gồm 4 thành phần: image, registry, engine, container



Thành phần của Docker	
Image	Là bao gồm các gói thực thi, biến môi trường,...-> Chạy được ứng dụng như sau khi tải sql server, mongoDB...
Container	Chứa image trong đó khi các ứng dụng trong image chạy.
Registry	Kho chứa image(public hoặc private)
Engine	Là phần mềm Docker

Các thao tác với docker

Trên_mạng Langbiang

- Kéo (tải) image về từ docker hub: docker pull tên_phần_mềm.
- Liệt kê các image hiện có: docker images

REPOSITORY: tên của image
TAG: phiên bản của image
IMAGE ID: mã định danh của image
CREATED: ngày tạo image
SIZE: kích thước của image

- Xóa image: docker rmi ten/id
- Liệt kê container đang chạy/tắt: docker ps/docker ps -a
- Xem và tải image bằng PowerShell
- Xóa một image trong docker
- Chuyển container thành image
- ...

PHP (HyperText Preprocessor)

Tổng quan

Có rất nhiều ngôn ngữ lập trình phía server như: C#, Java, Ruby, Python, PHP, JavaScript.
PHP là ngôn ngữ phía server, trong phần mở rộng của tập tin mã nguồn .php có thể chứa: văn bản, HTML, CSS, JavaScript, jQuery,...
Chú ý: Khi client yêu cầu 1 trang web nào đó, Webserver chuyển web mã nguồn PHP->HTML --->Gửi về cho client(Vì trình duyệt không hiểu và thông dịch được mã PHP)
Dấu ‘ và “ trong PHP:

- ‘: chuỗi trong nháy đơn sẽ không kiểm tra cú pháp nên ví dụ ‘\n’ thì chương trình không hiểu xuống dòng.
- “: chuỗi trong nháy kép được kiểm tra cú pháp.

Biến, hằng và toán tử

Biến: là vùng nhớ được đặt tên, tồn tại trong thời gian Server phát sinh trang web, sau đó nó sẽ được xóa bỏ.
Ví dụ: $bienne và dùng (.) để nối chuỗi, ví dụ $hoten=$ho.” “.$ten
Tên biến trong php phân biệt hoa và thường.

Hằng: Define(“Tèo”, 137)

Toán tử:

- So sánh
- Logic: and, or,...
- Điều kiện: if else,...

Kiểm soát luồng php

Phương thức Get và Post: là 2 phương thức của giao thức http, đều gửi dữ liệu về server xử lý khi người dùng nhấn submit trong form.

- Phương thức Get: dữ liệu sau khi submit thì bị hiển thị trên đường link
  + Không thể gửi đi các dữ liệu lớn: hình ảnh
  + Dữ liệu gửi đi được lưu lại trong query string, được webrowser cached ghi lại => lấy dữ liệu nhanh hơn post thì nếu có kết quả đã dc lưu thì trả về cho client ngay mà không cần xử lý tại serve.
- Phương thức Post: dữ liệu sau khi được gửi thì không bị lộ trên URL
  + Không giới hạn độ lớn dữ liệu
  + Có tính bảo mật hơn Get

Hàm sửa báo lỗi khi không xác định được biến sử dụng Post/ Get: Isset

```php
if( isset($_POST["so1"])&& isset($_POST["so2"])&& isset($_POST["pheptinh"]) ){
    $s1=$_POST["so1"];
    $s2=$_POST["so2"];
    $pheptinh=$_POST["pheptinh"];
```

Thư viện php

Bảo mật

Hướng đối tượng

Session và Cookie

Cookie: "Hiểu đơn giản như cái bánh quy " ghi chép, theo dõi " những hành động của người dùng khi duyệt web: *bấm vào nút nào, tìm kiếm gì,....*

- Mỗi trình duyệt trên máy tính sẽ có nơi lưu trữ cookie riêng.
- Có nhiều loại cookie: ...

Cơ chế hoạt động: khi **client** gửi yêu cầu xem trang web -> **gửi request(http get)** tới server chứa tên trang web và tên miền. ->Sau khi nhận yêu cầu thì **server** sẽ **gửi** về mã **trạng thái yêu cầu, cookie và nội dung trang web** :
Set-Cookie: name1=value1; attributes

Session: dùng để lưu giá trị biến giữa các trang web, ví dụ như tên và mật khẩu người dùng:

- Nội dung session chỉ được lưu tạm thời, bị xóa khi thoát khỏi website