Mục lục

[Kiến thức cần nắm 2](#_Toc531809095)

[MVC 2](#_Toc531809096)

[1. Tổ chức thư mục 2](#_Toc531809097)

[2. MODEL : Xử lý với Database 2](#_Toc531809098)

[2.1 Kết nối Cơ sở dữ liệu 2](#_Toc531809099)

[2.2 Tạo fuction INSERT/UPDATE/DELETE với Database và hàm trả về Id 2](#_Toc531809100)

[2.3 Tạo function SELECT một và nhiều dữ liệu trong database 3](#_Toc531809101)

[3. CONTROLLER: Xử lý load VIEW theo yêu cầu người dùng 4](#_Toc531809102)

[3.1 Tạo hàm loadView gọi tới file layout.view.php 4](#_Toc531809103)

[4. VIEW: tạo file Layout.view.php để gọi tới các trang view hiển thị 4](#_Toc531809104)

[4.1 Tạo trang layout.view.php 4](#_Toc531809105)

[4.2 Tạo các trang nội dung của web với định dạng index.view.php 4](#_Toc531809106)

[5. FIRST LOAD: load trang index.php 4](#_Toc531809107)

[5.1 Cách dữ liệu truyền trong mô hình MVC 4](#_Toc531809108)

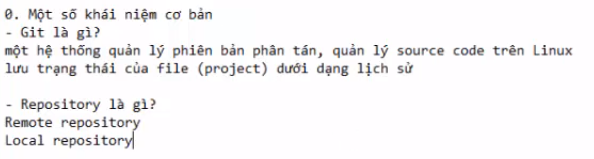
[5.2 Tạo VIEW/layout.view.php để lưu HEADER, MENU, FOOTER 5](#_Toc531809109)

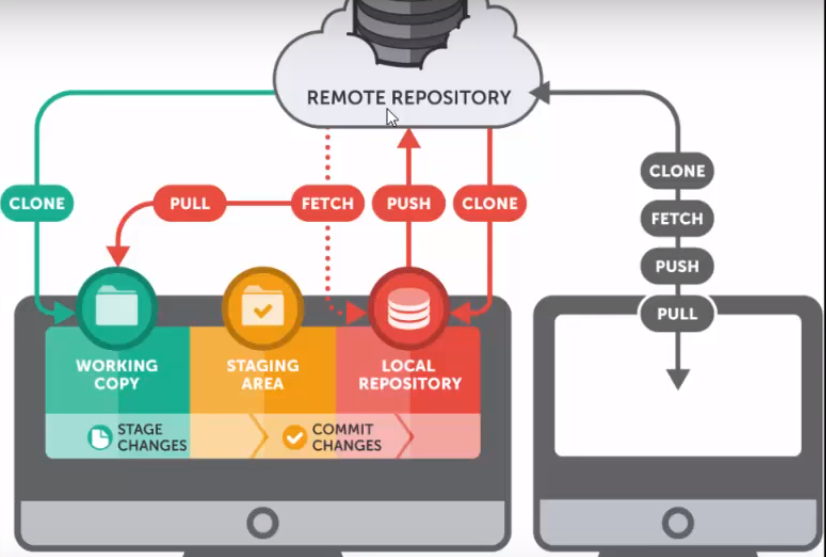
[5.3 Tạo VIEW/index.view.php 5](#_Toc531809110)

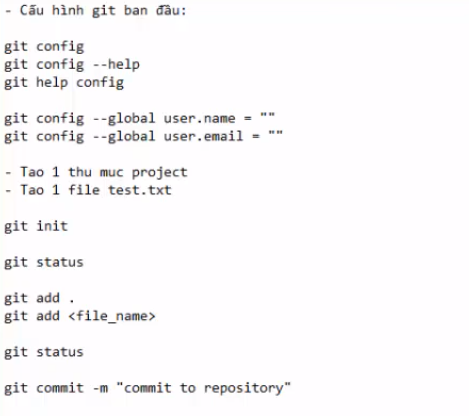
[5.4 Tạo CONTROLLER/indexController.php kế thừa class BaseController để load trang index 5](#_Toc531809111)

# Kiến thức cần nắm

## Git







**Kiểm tra cài đặt git**

Sau khi cài đặt: vào CMD kiểm tra bằng lệnh

git –version

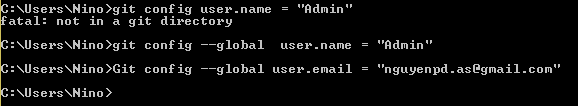
**Cấu hình git**

Tên user

Git config –global user.name = “NinoPham2303”

Tên user

Git config –global user.email = [nguyenpd.as@gmail.com](mailto:nguyenpd.as@gmail.com)



**Tạo folder cần cho lên git**

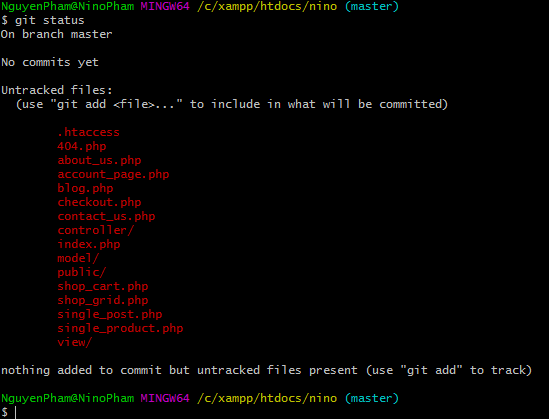
Bên trong folder cần cho lên Git ta chuột phải và git base here và gõ lệnh

Git init



**Xem trạng thái file trong folder**

Git status



**Git xác nhận các file trên, dùng lệnh**

**Cách 1:**

//Git từng file

Git add “tenfile”

// Git tất cả

Git add \*

Git add .

Git commit –m “noi dung sua doi”

**Cách 2(viết tắt) :**

Git commit –a –m ’Noi dung thay doi’

**Push lên Github**

Tạo folder trên trạng git của bạn rồi copy đường dẫn folder đó, sau đó push lên:

Git remote add ten\_folder đường\_dẫn\_folder

Git push –set-upstream ten\_folder master

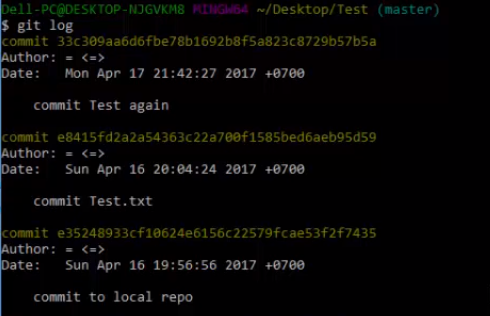
**Lấy project trên github về**

Git clone đường\_dẫn\_url\_project

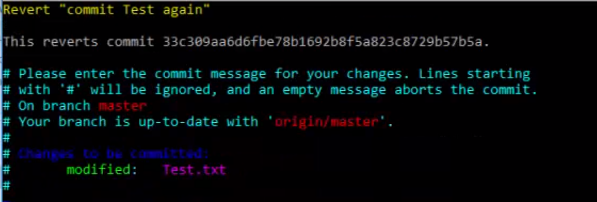
**Backup dữ liệu trên github**

Kiểm tra các file đã update lên git và revert nó về

Git log



Git revert dãy\_số\_commit\_cần\_revert



**CTRL c và :q để thoát**

**Update project về khi làm việc nhóm**

Git pull

# MVC

# Tổ chức thư mục

CONTROLLER: để các thư mục thao tác với Model và View

MODEL: Để thao tác với CSDL

VIEW: chứa các nội dung trang hiển thị.

PUBLIC: chứa các trang html

# MODEL : Xử lý với Database

## Kết nối Cơ sở dữ liệu

Trong thư mục Model tạo class DBConnect để kết nối CSDL.

/\*\*

\* Connect Database

\*/

*class* DBConnect

{

/\*\*

\* @var *PDO* $Connect

\*/

private $connect;

/\*\*

\* Connect to database

\*/

public *function* \_\_construct($dbname = 'shop2509',$user,$pass)

{

$this->connect = new *PDO*("mysql:host=localhost;dbname=$dbname",$user,$pass);

$this->connect->exec('set names utf8');

}

## Tạo fuction INSERT/UPDATE/DELETE với Database và hàm trả về Id

/\*\*

\* @var *PDOStatement* $statement

\*/

private $statement;

/\*\*

\* Use for Insert/Update/Delete

\* @param string $sql

\* @param array $data = []

\* @return bool

\*/

*function* executeQuery(*string* $sql, *array* $data = []){

$this->statement = $this->connect->prepare($sql);

for ($i = 1; $i < count($data); $i++) {

$this->statement->bindParam($i,$data[$i-1]);

}

return $this->statement->execute();

}

/\*\*

\* Get recent insert Id

\* @return int

\*/

*function* getInsertedId(){

return $this->connect->lastInsertId();

}

## Tạo function SELECT một và nhiều dữ liệu trong database

/\*\*

\* Use for SELECT query, return only 1 row in table

\* @param string $sql

\* @param array $data

\* @return object|false

\*/

*function* loadOneRow(*string* $sql, *array* $data = []){

$check = $this->executeQuery($sql, $data);

return ($check) ? $this->statement->fetch(*PDO*::FETCH\_OBJ) : false;

}

/\*\*

\* Use for SELECT query, return more rows in table

\* @param string $sql

\* @param array $data

\* @return object|false

\*/

*function* loadMoreRow(*string* $sql, *array* $data = []){

$check = $this->executeQuery($sql, $data);

return ($check) ? $this->statement->fetchAll(*PDO*::FETCH\_OBJ) : false;

}

# CONTROLLER: Xử lý load VIEW theo yêu cầu người dùng

## Tạo hàm loadView gọi tới file layout.view.php

*class* BaseController

{

/\*\*

\* Gọi view từ layout

\* @param string $view

\* @param array $data

\*/

public *function* loadView(*string* $view = 'index',*array* $data = [])

{

include\_once 'view/layout.view.php';

}

}

# VIEW: tạo file Layout.view.php để gọi tới các trang view hiển thị

## Tạo trang layout.view.php

Trong file Layout.view.php, ta sẽ cố định header và footer của trang web, đồng thời nội dung sẽ gọi tớ bằng Include\_once ‘$view.view.php’ để xử lý yêu cầu load trang của người dùng.

<div>HEADER (cố định)</div>

<div>MENU (cố định)</div>

<?php include\_once "$view.view.php"; ?>

<div>FOOTER (cố định)</div>

## Tạo các trang nội dung của web với định dạng index.view.php

# FIRST LOAD: load trang index.php

Index.php 🡪 getHomepage() / IndexController

🡪 loadView(‘index’) / BaseController 🡪 include\_one(layout.view.php)

🡪include\_once($view.view.php) với $view = index.

## Cách dữ liệu truyền trong mô hình MVC

**2**

Tạo file indexModel trong MODEL để chứa các function load tất cả dữ liệu cho trang Index

Tạo file nội dung hiển thị index.view.php trong VIEW

**1**

Tạo file IndexController trong CONTROLLER để load trang index.view.php ra trình duyệt

Tạo file IndexController trong CONTROLLER để load trang index.view.php ra trình duyệt

Ở trang index thì gọi hàm để hiện thị

## Tạo VIEW/layout.view.php để lưu HEADER, MENU, FOOTER

## Tạo VIEW/index.view.php

Trong file này chứa HTML nội dung để hiển thị.

## Tạo CONTROLLER/indexController.php kế thừa class BaseController để load trang index

require\_once 'BaseController.php';

/\*\*

\* Load dữ liệu trang Index

\*/

*class* IndexController extends *BaseController*

{

/\*\*

\* load trang index

\*/

public *function* getHompage()

{

return $this->loadView('index');

}

}

Tạo MODEL