**Thư mục config -> config.php**

<?php

define("DB\_HOST", "localhost");

define("DB\_USER", "root");

define("DB\_PASS", "");

define("DB\_NAME", "your\_db\_name");

?>

**Thư mục helpers-> format.php**

<?php

/\*\*

\* Format Class

\*/

class Format{

public function formatDate($date){

return date('F j, Y, g:i a', strtotime($date));

}

public function textShorten($text, $limit = 400){

$text = $text. " ";

$text = substr($text, 0, $limit);

$text = substr($text, 0, strrpos($text, ' '));

$text = $text.".....";

return $text;

}

public function validation($data){

$data = trim($data);

$data = stripcslashes($data);

$data = htmlspecialchars($data);

return $data;

}

public function title(){

$path = $\_SERVER['SCRIPT\_FILENAME'];

$title = basename($path, '.php');

//$title = str\_replace('\_', ' ', $title);

if ($title == 'index') {

$title = 'home';

}elseif ($title == 'contact') {

$title = 'contact';

}

return $title = ucfirst($title);

}

}

?>

**Thư mục lib -> database.php**

<?php

Class Database{

public $host = DB\_HOST;

public $user = DB\_USER;

public $pass = DB\_PASS;

public $dbname = DB\_NAME;

public $link;

public $error;

public function \_\_construct(){

$this->connectDB();

}

private function connectDB(){

$this->link = new mysqli($this->host, $this->user, $this->pass,

$this->dbname);

if(!$this->link){

$this->error ="Connection fail".$this->link->connect\_error;

return false;

}

}

// Select or Read data

public function select($query){

$result = $this->link->query($query) or

die($this->link->error.\_\_LINE\_\_);

if($result->num\_rows > 0){

return $result;

} else {

return false;

}

}

// Insert data

public function insert($query){

$insert\_row = $this->link->query($query) or

die($this->link->error.\_\_LINE\_\_);

if($insert\_row){

return $insert\_row;

} else {

return false;

}

}

// Update data

public function update($query){

$update\_row = $this->link->query($query) or

die($this->link->error.\_\_LINE\_\_);

if($update\_row){

return $update\_row;

} else {

return false;

}

}

// Delete data

public function delete($query){

$delete\_row = $this->link->query($query) or

die($this->link->error.\_\_LINE\_\_);

if($delete\_row){

return $delete\_row;

} else {

return false;

}

}

}

**Thư mục lib -> session.php**

<?php

/\*\*

\*Session Class

\*\*/

class Session{

public static function init(){

if (version\_compare(phpversion(), '5.4.0', '<')) {

if (session\_id() == '') {

session\_start();

}

} else {

if (session\_status() == PHP\_SESSION\_NONE) {

session\_start();

}

}

}

public static function set($key, $val){

$\_SESSION[$key] = $val;

}

public static function get($key){

if (isset($\_SESSION[$key])) {

return $\_SESSION[$key];

} else {

return false;

}

}

public static function checkSession(){

self::init();

if (self::get("login")== false) {

self::destroy();

header("Location:login.php");

}

}

public static function checkLogin(){

self::init();

if (self::get("login")== true) {

header("Location:index.php");

}

}

public static function destroy(){

session\_destroy();

header("Location:login.php");

}

}

?>