FPT Education

FPT POLYTECHNIC

cdio™

Conceive Design Implement Operate

THỰC HỌC – THỰC NGHIỆP

LẬP TRÌNH PHP2

PHP OOP VỚI DATABASE

http://www.poly.edu.vn

Mục tiêu

- Login & Regist system (tiếp tục)
  - Login sử dụng PDO
  - Regist
  - Regist sử dụng PDO

PHẦN 1

LOGIN SYSTEM (CONT) – DATABASE USING PDO

```php
private PDO $pdo;

public function __construct(array $config)
{
    $defaultOptions = [
        PDO::ATTR_EMULATE_PREPARES   => false,
        PDO::ATTR_DEFAULT_FETCH_MODE => PDO::FETCH_ASSOC,
    ];

    try {
        $this->pdo = new PDO(
            $config['driver'] . ':host=' . $config['host'] . ';dbname=' . $config['database'],
            $config['user'],
            $config['pass'],
            $config['options'] ?? $defaultOptions
        );
    } catch (\PDOException $e) {
        throw new \PDOException($e->getMessage(), (int) $e->getCode());
    }
}

public function __call(string $name, array $arguments)
{
    return call_user_func_array([$this->pdo, $name], $arguments);
}
```

???

LOGIN SYSTEM (CONT) – DATABASE USING PDO

```php
private PDO $pdo;

public function __construct(array $config)
{
    $defaultOptions = [
        PDO::ATTR_EMULATE_PREPARES   => false,
        PDO::ATTR_DEFAULT_FETCH_MODE => PDO::FETCH_ASSOC,
    ];

    try {
        $this->pdo = new PDO(
            $config['driver'] . ':host=' . $config['host'] . ';dbname=' . $config['database'],
            $config['user'],
            $config['pass'],
            $config['options'] ?? $defaultOptions
        );
    } catch (\PDOException $e) {
        throw new \PDOException($e->getMessage(), (int) $e->getCode());
    }
}

public function __call(string $name, array $arguments)
{
    return call_user_func_array([$this->pdo, $name], $arguments);
}
```

From .env file HOW?

LOGIN SYSTEM (CONT) – DATABASE USING PDO – ENV FILE

mvc_simple > {} composer.json > ...

```
1   {
2       "nam
3       "aut
4           "psr-4": {
5               "app\\": "./"
6           }
7       },
8       "require": {
9           "vlucas/phpdotenv": "^5.4"
10      }
11  }
```

This is an object of namespaces (k
(values, can be arrays of paths) by

Install vlucas/phpdotenv

***Giảng viên có thể gợi ý các thư viện tương tự

LOGIN SYSTEM (CONT) – DATABASE USING PDO – ENV FILE

Index.php

```php
$dotenv = Dotenv\Dotenv::createImmutable(__DIR__);
$dotenv->load();
var_dump($_ENV);
```

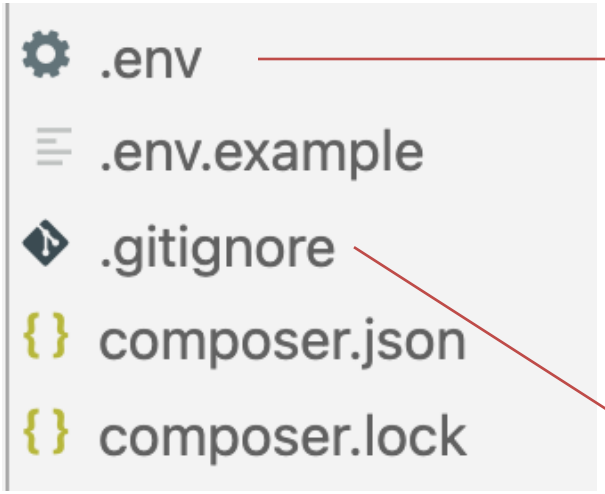
Class config

```php
class Config
{
    protected array $config = [];

    public function __construct(array $env)
    {
        $this->config = [
            'db' => [
                'host'     => $env['DB_HOST'],
                'user'     => $env['DB_USER'],
                'pass'     => $env['DB_PASS'],
                'database' => $env['DB_DATABASE'],
                'driver'   => $env['DB_DRIVER'] ?? 'mysql',
            ],
        ];
    }

    public function __get(string $name)
    {
        return $this->config[$name] ?? null;
    }
}
```

***Giảng viên có thể gợi ý các thư viện tương tự

LOGIN SYSTEM (CONT) – DATABASE USING PDO – ENV FILE

Cấu trúc



.env .gitignore

PHPMVCFRAMEWORK > .env

```
1   DB_DSN = mysql:host=192.168.64.2;dbname=mvc_framework
2   DB_USER = tram
3   DB_PASSWORD = ZMuOeX@F*F(xYN65
```

.env .gitignore

PHPMVCFRAMEWORK > .gitignore

```
1   vendor
2   .idea
3   .env
```

***Giảng viên có thể gợi ý các thư viện tương tự

LOGIN SYSTEM (CONT) – DATABASE USING PDO SELECT

- Chú ý (đây chỉ là gợi ý)

```php
$tableName = $this->tableName();
$attributes = array_keys($where);
$sql = implode("AND ", array_map(fn($attr) => "$attr = :$attr", $attributes));
$statement = self::prepare("SELECT * FROM $tableName WHERE $sql");
foreach($where as $key => $item){
    $statement->bindValue(":$key", $item);
}

$statement->execute();
return $statement->fetchObject(static::class);
//SELECT * FROM $tableName WHERE email= :email AND firstname = :firsname
```

LOGIN – PHP OOP VÀ PDO

PHẦN 2

LOGIN & REGIST SYSTEM (CONT)

```php
$router = new Router();
$router
  ->get('/', [app\Home::class,'index'])
  ->post('/upload', [app\Home::class,'upload'])
  ->get('/login', [app\Login::class,'login'])
  ->post('/login', [app\Login::class,'login'])
  ->get('/logout', [app\Login::class,'logout'])
  ->get('/register', [app\Register::class,'register'])
  ->post('/register', [app\Register::class,'register']);
```

???

url:
http://localhost:8080/register
request method: GET
action > class: Register >
method register function

LOGIN & REGIST SYSTEM (CONT)

```php
$router = new Router();
$router
  ->get('/', [app\Home::class,'index'])
  ->post('/upload', [app\Home::class,'upload'])
  ->get('/login', [app\Login::class,'login'])
  ->post('/login', [app\Login::class,'login'])
  ->get('/logout', [app\Login::class,'logout'])
  ->get('/register', [app\Register::class,'register'])
  ->post('/register', [app\Register::class,'register']);
```

???

Class: Register

```php
//view
return'<form action="/register" method="post">
    <div class="row">
    <div class="col">…
    </div>
    <div class="col">…
    </div>
    </div>
    <div class="form-group">
    <label>Email</label>
    <input type="text" class="form-control" name="email">
    </div>
    <div class="form-group">
    <label>Password</label>
    <input type="password" class="form-control" name="password">
    </div>
    <div class="form-group">
    <label>Confirm Password</label>
    <input type="password" class="form-control" name="confirmPassword">
    </div>
    <button type="submit" name="submit" class="btn btn-primary" value="Su
</form>';
```

LOGIN & REGIST SYSTEM (CONT)

```
$router = new Router();
$router
  ->get('/', [app\Home::class,'index'])
  ->post('/upload', [app\Home::class,'upload'])
  ->get('/login', [app\Login::class,'login'])
  ->post('/login', [app\Login::class,'login'])
  ->get('/logout', [app\Login::class,'logout'])
  ->get('/register', [app\Register::class,'register'])
  ->post('/register', [app\Register::class,'register']);
```

url:
http://localhost:8080/register
request method: POST
action > class: Register >
method register function

Class: Register

```
//view
return'<form action="/register" method="post">
    <div class="row">
    <div class="col">…
    </div>
    <div class="col">…
    </div>
    </div>
    <div class="form-group">
    <label>Email</label>
    <input type="text" class="form-control" name="email">
    </div>
    <div class="form-group">
    <label>Password</label>
    <input type="password" class="form-control" name="password">
    </div>
    <div class="form-group">
    <label>Confirm Password</label>
    <input type="password" class="form-control" name="confirmPassword">
    </div>
    <button type="submit" name="submit" class="btn btn-primary" value="Su
</form>';
```

LOGIN & REGIST SYSTEM (CONT)

```php
$router = new Router();
$router
  ->get('/', [app\Home::class,'index'])
  ->post('/upload', [app\Home::class,'upload'])
  ->get('/login', [app\Login::class,'login'])
  ->post('/login', [app\Login::class,'login'])
  ->get('/logout', [app\Login::class,'logout'])
  ->get('/register', [app\Register::class,'register'])
  ->post('/register', [app\Register::class,'register']);
```

```php
if(isset($_POST['submit'])){
    $this->fname = $_POST['firstname'];
    $this->lname = $_POST['lastname'];
    $this->pwd = $_POST['password'];
    $this->pwdRepeat = $_POST['confirmPassword'];
    $this->email = $_POST['email'];

    $this->signupUser();
    header("location:/");
}
```

Class: Register
Register method

Class: Register
Register method

```php
//view
return'<form action="/register" method="post">
    <div class="row">
    <div class="col">…
    </div>
    <div class="col">…
    </div>
    </div>
    <div class="form-group">
    <label>Email</label>
    <input type="text" class="form-control" name="email">
    </div>
    <div class="form-group">
    <label>Password</label>
    <input type="password" class="form-control" name="password">
    </div>
    <div class="form-group">
    <label>Confirm Password</label>
    <input type="password" class="form-control" name="confirmPassword">
    </div>
    <button type="submit" name="submit" class="btn btn-primary" value="Su
</form>';
```

LOGIN & REGIST SYSTEM (CONT)

```php
protected function signupUser(){

    if($this->emptyInput()==false){
        header("location:/register?error=emptyInput");
        exit();
    }
    if($this->invalidEmail()==false){
        header("location:/register?error=email");
        exit();
    }
    if($this->pwdMatch()==false){
        header("location:/register?error=passwordMatch");
        exit();
    }
    if($this->emptyInput()==false){
        header("location:/register?error=emptyInput");
        exit();
    }

    $this->setUser($this->fname, $this->lname, $this->pwd, $this->email);
}
```

signupUser

emptyInput,
invalidEmail,
pwdMatch

Controller
hay Model?

LOGIN & REGIST SYSTEM (CONT)

```php
protected function setUser($fname, $lname, $pwd, $email){
    $status = 0;
    $now = date("Y-m-d H:i:s");
    $hashedPwd = password_hash($pwd, PASSWORD_DEFAULT);

    $params = [$email, $fname, $lname, $status, $now, $hashedPwd];
    $INSERT = "INSERT INTO `users`(`email`, `firstname`, `lastname`, `status`, `created_at`, `password`) VALUES
    ('{$email}','{$fname}','{$lname}',{$status},'{$now}','{$hashedPwd}')";
    $stmt = $this->connect()->query($INSERT);

    $resultCheck = '';
    if ($stmt === FALSE) {
        $resultCheck = false;
        $stmt = null;
        header("location:register?error=stmtfailed");
        exit();
    } else {
        $resultCheck = true;
    }

    return $resultCheck;
}
```

setUser: Controller hay Model?

LOGIN & REGIST SYSTEM (CONT)

- Gợi ý các validate (method nào là controller, model)

```php
public function emptyInput(){
    $result = null;
    if(empty($this->fname)||empty($this->lname)||empty($this->pwd)
    ||empty($this->pwdRepeat)||empty($this->email)){
        $result = false;
    } else {
        $result = true;
    }
    return $result;
}
public function invalidEmail(){
    $result = null;
    if(!filter_var($this->email, FILTER_VALIDATE_EMAIL)){
        $result = false;
    } else {
        $result = true;
    }
    return $result;
}
```

LOGIN & REGIST SYSTEM (CONT)

- Gợi ý các validate (method nào là controller, model)

```php
public function pwdMatch(){
    $result = null;
    if($this->pwd !== $this->pwdRepeat){
        $result = false;
    } else {
        $result = true;
    }
    return $result;
}

public function emailTakenCheck(){
    $result = false;
    if($this->checkUser($this->email)){
        $result = false;
    } else {
        $result = true;
    }
    return $result;
}
```

LOGIN & REGIST SYSTEM (CONT)

- Gợi ý các validate (method nào là controller, model)

```php
protected function checkUser($email){
    $stmt = $this->connect()->prepare('SELECT * FROM users WHERE email=?');
    if(!$stmt->execute($email)){
        $stmt = null;
        header("location:register?error=stmtfailed");
        exit();
    }
    $stmt->close();
    $resultCheck = true;
    if($stmt->num_rows > 0){
        $resultCheck = false;
    }else{
        $resultCheck = true;
    }
    return $resultCheck;
}
```

REGIST

LOGIN SYSTEM (CONT) – DATABASE USING PDO INSERT

- Chú ý (đây chỉ là gợi ý)

```
$tableName = $this->tableName();
$attributes = $this->attributes();
$params = array_map(fn($attr) => ":$attr", $attributes);
$statement = self::prepare("INSERT INTO $tableName (".implode(',', $attributes).")
    VALUES(".implode(',',$params).")");
//var_dump($statement, $params, $attributes);
foreach($attributes as $attribute)
{
    $statement->bindValue(":$attribute", $this->{$attribute});
}
$statement->execute();
return true;
```

REGIST – PHP OOP VÀ PDO

SUMARRY

- ☑ Login & Regist system (tiếp tục)
  - ❖ Login sử dụng PDO
  - ❖ Regist
  - ❖ Regist sử dụng PDO


thank
you!