

My Phonebook

Xây dựng ứng dụng web **My Phonebook** hỗ trợ các chức năng sau:

- Chứng thực người dùng (đăng ký, đăng nhập)
- Sau khi đăng nhập, người dùng có thể xem, thêm, sửa, xóa các contact của mình. Thông tin mỗi contact gồm có tên (name), số điện thoại (phone number) và ghi chú (notes)

Cài đặt, cấu hình ứng dụng:

1. Tạo CSDL *phonebook* và import script *phonebook.sql* đã cho CSDL này
2. Sao chép thư mục *phonebook* đã cho vào một vị trí nào đó trên máy, giả sử tại thư mục *C:/xampp/apps*
3. Cấu hình host ảo cho thư mục *C:/xampp/apps/phonebook/public*

i. Hiệu chỉnh tập tin *C:/xampp/apache/conf/httpd.conf*.

```
4. ...  
5. Listen 80  
6. Listen 8080  
7. ...
```

ii. Hiệu chỉnh tập tin *C:/xampp/apache/conf/extra/httpd-vhosts.conf*.

```
8. <VirtualHost *:8080>  
9.     DocumentRoot "C:/xampp/apps/phonebook/public"  
10.    ServerName localhost  
11.    # Set access permission  
12.    <Directory "C:/xampp/apps/phonebook/public">  
13.        AllowOverride None  
14.        Require all granted  
15.  
16.        RewriteEngine On  
17.        RewriteCond %{REQUEST_FILENAME} !-f  
18.        RewriteCond %{REQUEST_FILENAME} !-d  
19.        RewriteRule ^.*$ index.php [L]  
20.    </Directory>  
21. </VirtualHost>
```

iii. Restart lại server Apache

22. Vào thư mục gốc của dự án (thư mục *phonebook*), mở tập tin *db.php* để thay đổi các thông số kết nối đến CSDL

23. Mở Command Prompt và thực thi lệnh `composer update` để tải về các thư viện phụ thuộc

24. Kiểm tra chức năng đăng ký (<http://localhost:8080/register>), đăng nhập (<http://localhost:8080/login>) hoạt động

Tìm hiểu mã nguồn dự án:

Cấu trúc mã nguồn dự án **My Phonebook**:

```
phonebook
+---app                                // Namespace bắt đầu bằng App
+---Controllers                       // Namespace bắt đầu bằng App\Controllers
+---Auth                             // Namespace bắt đầu bằng App\Controllers\Auth
+---LoginController.php
+---RegisterController.php
+---Controller.php
+---HomeController.php
+---Models                           // Namespace bắt đầu bằng App\Models
+---User.php
+---Router.php
+---SessionGuard.php
+---public
+---css
+---fonts
+---js
+---index.php                        // Front Controller, nơi định nghĩa các route
+---vendor
+---views                            // Chứa các tập tin định nghĩa các 'views' cho ứng dụng
+---auth
+---login.php
+---register.php
+---layouts
+---default.php
+---home.php
+---composer.json
+---composer.lock
+---db.php                           // Thông số kết nối đến CSDL (Eloquent)
+---helpers.php                      // Các hàm trợ giúp
+---server.php                       // Chạy ứng dụng với PHP built-in server
```

Một số hàm đã được định nghĩa sẵn:

- Chuyển hướng đến một URL khác: `redirect($url)`
- Đọc ra VÀ xóa một phần tử trong mảng `$_SESSION`: `session_get_once($key)`
- Kiểm tra người dùng đã đăng nhập hay chưa?: `\App\SessionGuard::checkLogin()`
- Lấy thông tin người dùng đã đăng nhập: `\App\SessionGuard::user()`
- Load một tập tin nào đó nằm trong thư mục *views*, ví dụ:

```
public function showLoginForm()
{
    ...
    $data = [...]; // Dữ liệu sẽ gửi đến script views/auth/login.php
```

```
// Load tập tin views/auth/login.php
$loginView = $this->view->render('auth/login', $data);
echo $loginView;
}
```

Các routes đã định nghĩa (trong tập tin *index.php*):

```
POST    /logout    \App\Controllers\Auth\LoginController@logout
GET     /register   \App\Controllers\Auth\RegisterController@showRegisterForm
POST    /register   \App\Controllers\Auth\RegisterController@register
GET     /login      \App\Controllers\Auth\LoginController@showLoginForm
POST    /login      \App\Controllers\Auth\LoginController@login

GET     /           \App\Controllers\HomeController@index
GET     /home     \App\Controllers\HomeController@index
```

Điều gì xảy ra khi người dùng truy xuất đến một route không tồn tại?

Tạo trang thông báo lỗi:####

1. Trong thư mục *views*, tạo thư mục *errors*. Tạo tập tin *errors/404.php* với nội dung lấy từ *errors_404.txt* đã cho

```
<?php $this->layout("layouts/default", ["title" => APPNAME]) ?>

<?php $this->start("page_specific_css") ?>
    <style>
        .error-template {padding: 40px 15px;text-align: center;}
        .error-actions {margin-top:15px;margin-bottom:15px;}
        .error-actions .btn { margin-right:10px; }
    </style>
<?php $this->stop() ?>

<?php $this->start("page") ?>
<div class="container">
    <div class="row">
        <div class="col-md-12">
            <div class="error-template">
                <h1>Page Not Found <small>
                    <font face="Tahoma" color="red">Error 404</font></small></h1>
                <div class="error-details">
                    Sorry, an error has occurred.
                    The page you requested could not be found.
                </div>
                <div class="error-actions">
                    <a href="/" class="btn btn-primary btn-lg">
                        <span class="glyphicon glyphicon-home"></span>
                        Take Me Home </a>
                </div>
            </div>
        </div>
    </div>
</div>
<?php $this->stop() ?>
```

2. Mở tập tin *index.php*, thêm lệnh sau ngay trước lệnh `Router::dispatch()`

```
Router::error('\App\Controllers\Controller@notFound');
```

3. Mở tập tin *app/Controllers/Controller.php*, thêm định nghĩa hàm `notFound`

```
public function notFound()
{
    http_response_code(404);
    echo $this->view->render('errors/404');
    exit();
}
```

4. Dùng trình duyệt kiểm tra lại rằng khi người dùng truy xuất đến route không tồn tại thì trang thông báo lỗi vừa tạo sẽ được hiển thị