

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)

Yêu cầu: File báo cáo cần nộp là file PDF chứa hình ảnh minh họa kết quả thực hiện. Trong file báo cáo ghi rõ thông mã sinh viên và họ tên sinh viên

Cấu hình MySQL và Apache:

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*

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

```
<VirtualHost *:80>
    DocumentRoot "C:/xampp/apps/phonebook/public"
    ServerName ct275-lab5.localhost
    # Set access permission
    <Directory "C:/xampp/apps/phonebook/public">
        AllowOverride None
        Require all granted

        RewriteEngine On
        RewriteCond %{REQUEST_FILENAME} !-f
        RewriteCond %{REQUEST_FILENAME} !-d
        RewriteRule ^.*$ index.php [L]
    </Directory>
</VirtualHost>
```

2. Restart lại server Apache
4. 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
5. Mở Command Prompt/Windows Terminal và thực thi lệnh `composer update` để tải về các thư viện phụ thuộc
6. Kiểm tra chức năng đăng ký (<http://ct275-lab5.localhost/register>), đăng nhập (<http://ct275-lab5.localhost/login>)

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
+---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: `exit_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::isUserLoggedIn()`
- 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
    $this->sendPage('auth/login', $data);
}

```

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() ?>
```

Trong dự án này, chúng ta sử dụng gói thư viện *league/plates* (<https://packagist.org/packages/league/plates>) để tạo các view cho ứng dụng. *league/plates* cho phép định nghĩa các trang web theo một layout. Bên trong layout, chúng ta có thể định nghĩa các section (ví dụ `<?=$this->section("page")?>`). Nội dung của các section trong layout sẽ được định nghĩa chi tiết bởi các trang sử dụng layout đó. Trong trang 404 vừa tạo, chúng ta chỉ định layout sẽ dùng (*views/layouts/default.php*) và nội dung chi tiết cho 2 section là *page_specific_css* và *page*

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

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

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

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

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