**TRƯỜNG CAO ĐẲNG CÔNG NGHỆ THỦ ĐỨC**

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO**

**LẬP TRÌNH BACK-END WEB 2**

**Nhóm A:**

|  |  |
| --- | --- |
| Hồ Sĩ Hùng | 19211TT0047 |
| Nguyễn Phúc Linh | 19211TT0608 |
| Nguyễn Quốc Phương | 19211TT0015 |
| Phan Tài Đức | 19211TT1501 |
| Nguyễn Quốc Việt | 19211TT1148 |

**MỤC LỤC**

[Log in 1](#_heading=h.gjdgxs)

[Tổng quan 1](#_heading=h.30j0zll)

[Phân tích code 1](#_heading=h.1fob9te)

[Log out 3](#_heading=h.3znysh7)

[Tổng quan 3](#_heading=h.2et92p0)

[Phân tích code 4](#_heading=h.tyjcwt)

[Sign Up 4](#_heading=h.3dy6vkm)

[Tổng quan 4](#_heading=h.1t3h5sf)

[Phân tích code 8](#_heading=h.4d34og8)

[Forgot password 13](#_heading=h.2s8eyo1)

[Tổng quan 13](#_heading=h.17dp8vu)

[Phân tích code 15](#_heading=h.3rdcrjn)

[Các trường hợp ngoại lệ 19](#_heading=h.26in1rg)

[View List User 20](#_heading=h.lnxbz9)

[Tổng quan 20](#_heading=h.35nkun2)

[Phân tích code 21](#_heading=h.1ksv4uv)

[Sort List 25](#_heading=h.44sinio)

[Tổng quan 25](#_heading=h.2jxsxqh)

[Phân tích code 26](#_heading=h.z337ya)

[Search User 27](#_heading=h.3j2qqm3)

[Tổng quan 27](#_heading=h.1y810tw)

[Phân tích code 28](#_heading=h.4i7ojhp)

[View User Infomation 29](#_heading=h.2xcytpi)

[Tổng quan 29](#_heading=h.1ci93xb)

[Phân tích code 31](#_heading=h.3whwml4)

[Add User (Admin) 32](#_heading=h.2bn6wsx)

[Tổng quan, phân tích code 33](#_heading=h.qsh70q)

[Các trường hợp ngoại lệ 36](#_heading=h.1pxezwc)

[Edit User (Admin) 39](#_heading=h.49x2ik5)

[Giao diện 39](#_heading=h.2p2csry)

[Phân tích code 41](#_heading=h.147n2zr)

[Các trường hợp ngoại lệ 45](#_heading=h.3o7alnk)

[Delete User 46](#_heading=h.23ckvvd)

[Tổng quan 46](#_heading=h.ihv636)

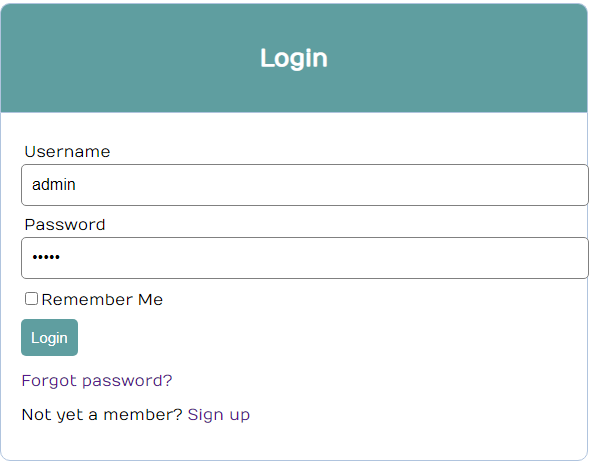
[Phân tích code 47](#_heading=h.32hioqz)

# **Log in**

## Tổng quan

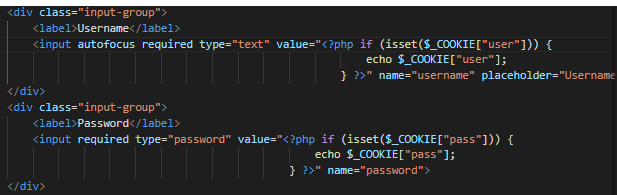
**Trường hợp nhập đúng username và password:**

Đăng nhập giữa Admin và user là hoàn toàn giống nhau chỉ cần thực hiện điền đúng username và password nhấn nút button “**Login**” để đăng nhập vào hệ thống.

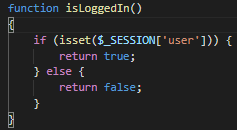


## Phân tích code

Code thực hiện kiểm tra đầu vào, hệ thống sẽ thực hiện kiểm tra cookie user và pass kiểm tra đúng thì sẽ đăng nhập vào hệ thống thành công.

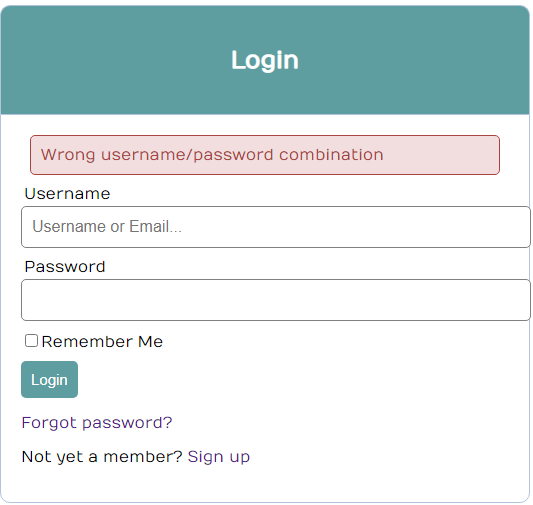


Hàm kiểm tra username đăng nhập.

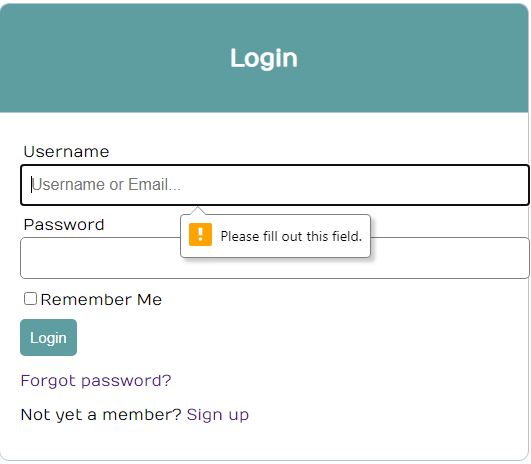


**Trường hợp nhập sai username hoặc password:**

Khi người dùng nhập sai username hoặc là password hệ thống sẽ thông báo “username hoặc là password sai”.



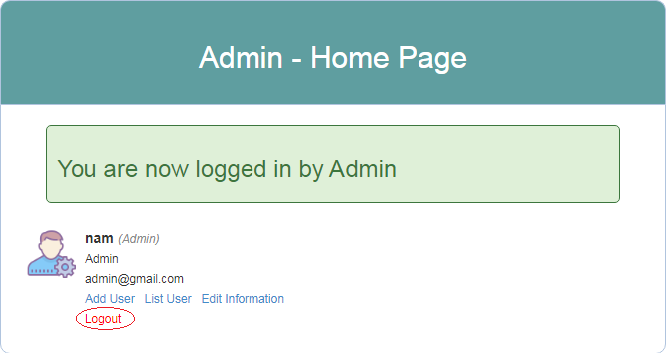
Trường hợp người dùng khi đăng nhập nhưng không điền username và password nhưng lại bấm “Login” thì hệ thống sẽ thông báo “vui lòng nhập thông tin vào các trường dữ liệu”, nhờ thuộc tính “required” trong code thực hiện chức năng này.



# **Log out**

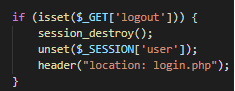
## Tổng quan

- Khi đã đăng nhập thành công vào hệ thống thành công nếu bạn muốn đăng xuất tài khoản thì bạn cần thực hiện nhấn vào nút “**Logout**” trên màn hình để thực hiện đăng xuất tài khoản.



## Phân tích code

+ Hệ thống sẽ kiểm tra khi logout đồng thời sẽ phá huỷ tất cả dữ liệu được liên kết với phiên hiện tại, sau đó thực hiện loại bỏ user và trở về trang login đăng nhập ban đầu:



# Sign Up

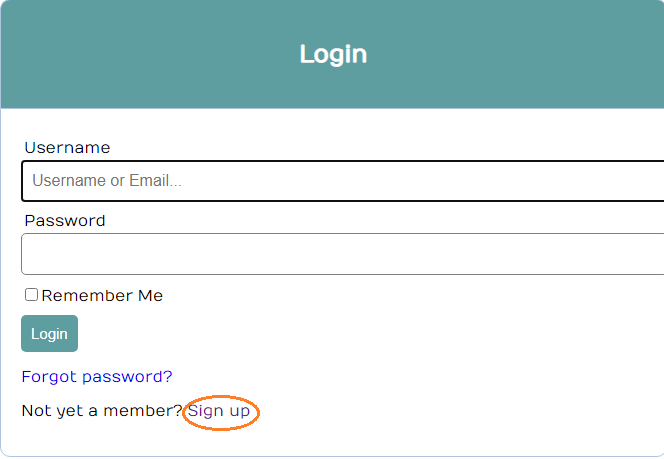
## Tổng quan

- Thực hiện truy cập vào web và form login sẽ hiện lên:

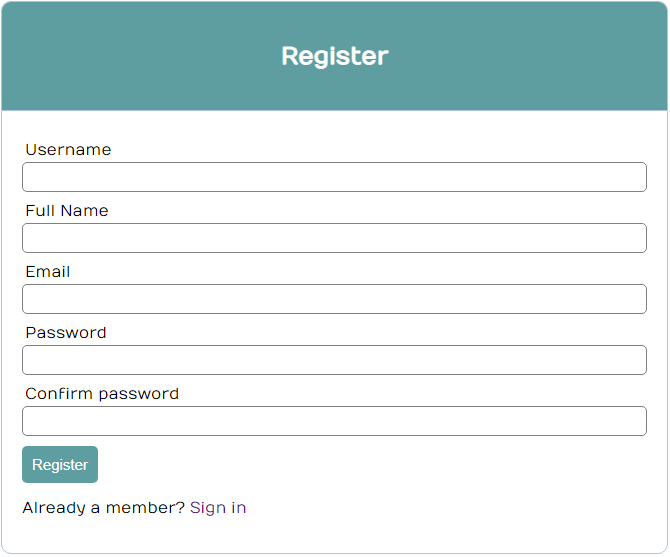
+ Khi bạn chưa có tài khoản thì bạn cần phải thực hiện đăng ký tài khoản để thực hiện đăng nhập:

Thực hiện nhấn chọn “Sign up”:

+ Sau khi nhấn chọn “Sign up” thì sẽ hiển thị đến form đăng ký:



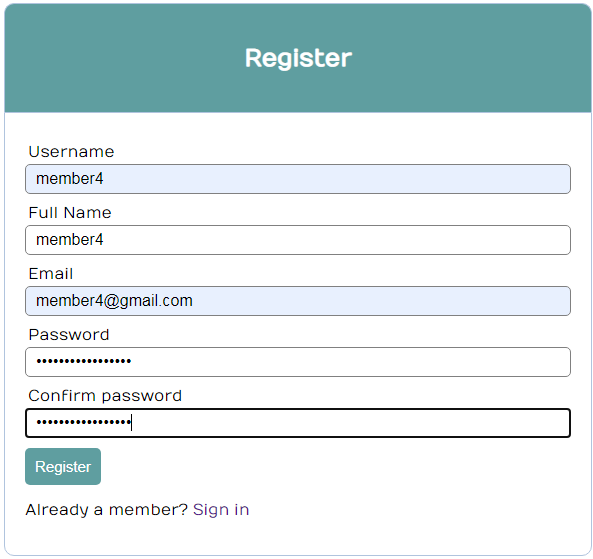
* Form đăng ký tài khoản:



* Thực hiện điền thông tin để đăng ký tài khoản:

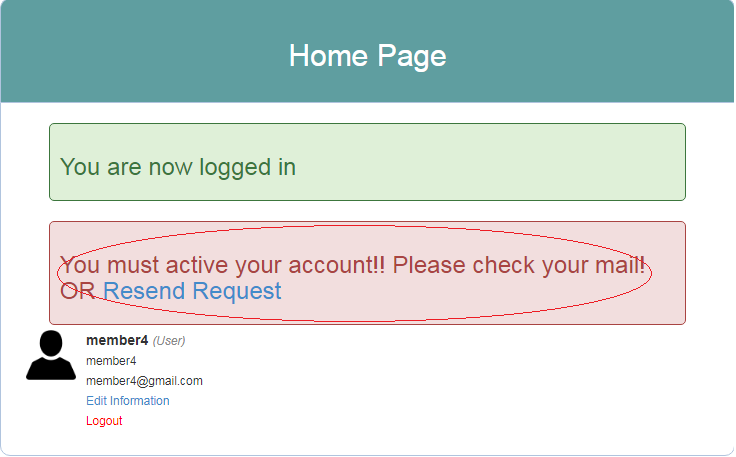
+ Sau khi đã điền đầy đủ thông tin thì thực hiện nhấn nút “**Register**” để thực hiện đăng ký tài khoản thành công. Trường hợp đăng ký tài khoản thành công gồm:

* Các trường dữ liệu không được để trống.
* Không bị trùng username người dùng với cơ sở dữ liệu.
* Không được trùng Email.
* Password phải đủ 8 ký tự trở lên phải có chữ số, bao gồm chữ in hoa, chữ thường và chứa ký tự đặc biệt.
* Confirm password phải nhập giống password để xác thực.



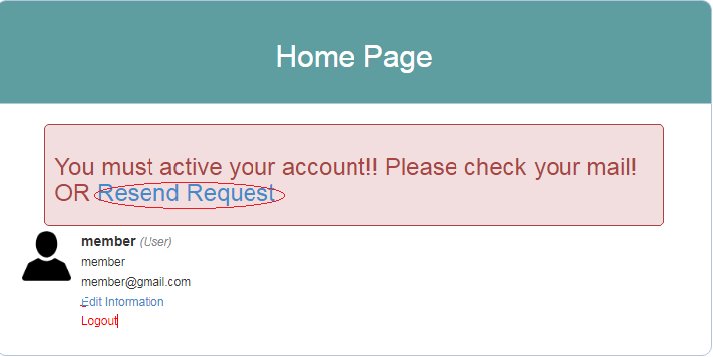
* Sau khi thực hiện đăng ký thành công hệ thống sẽ bắt buộc xác thực email:

+ Khi người dùng đăng ký thành công tài khoản thì hệ thống sẽ yêu cầu người dùng xác thực email.



* Nhấn chọn “**Resend Request**” để hệ thống gửi email xác thực đến người dùng:

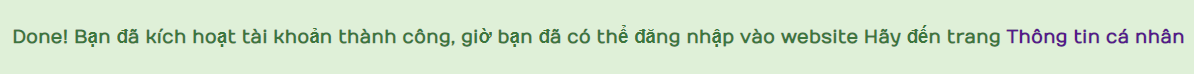
+ Trường hợp người dùng không xác nhận email của hệ thống thì sau khi đăng nhập tài khoản thì hệ thống vẫn thông báo để người dùng thực hiện xác nhận email.



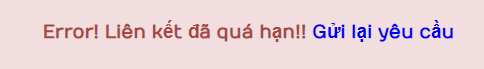
* Sau đó sẽ có email xác nhận đến email người dùng:



* Sau khi người dùng nhấn vào button “**VERIFY ACCOUNT**” thì tài khoản của người dùng đã được xác thực email và đăng nhập bình thường.

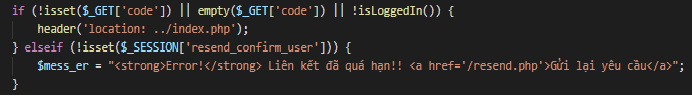


* Trường hợp người dùng không nhấn vào button “**VERIFY ACCOUNT**” nếu chờ quá lâu hệ thống sẽ thông báo.



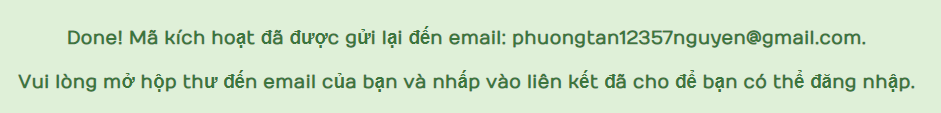
## Phân tích code

+ Hệ thống sẽ kiểm tra nếu người không nhấn vào liên kết để thực hiện xác nhận email thì hệ thống sẽ báo trễ hạn và cần phải xác nhận gửi lại yêu cầu để xác thực email thêm lần nữa.

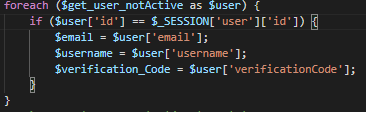


* Khi người dùng đã xác nhận lại yêu cầu.

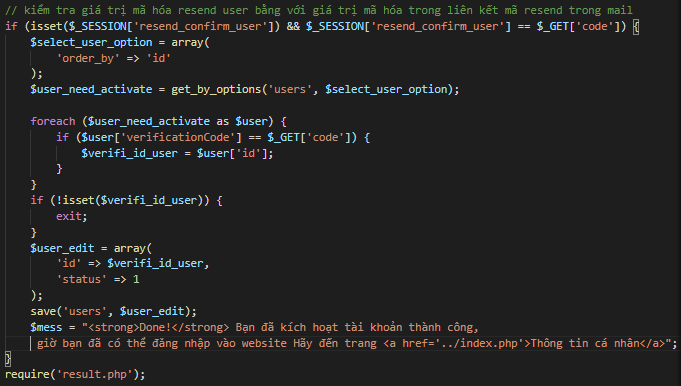
+ Sau đó người dùng xác nhận lại yêu cầu thì email sẽ được kích hoạt lại lần nữa.



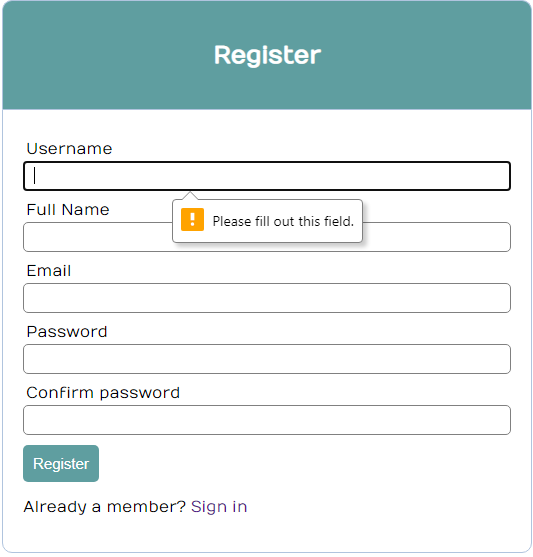
* Code thực hiện xác minh email người dùng:



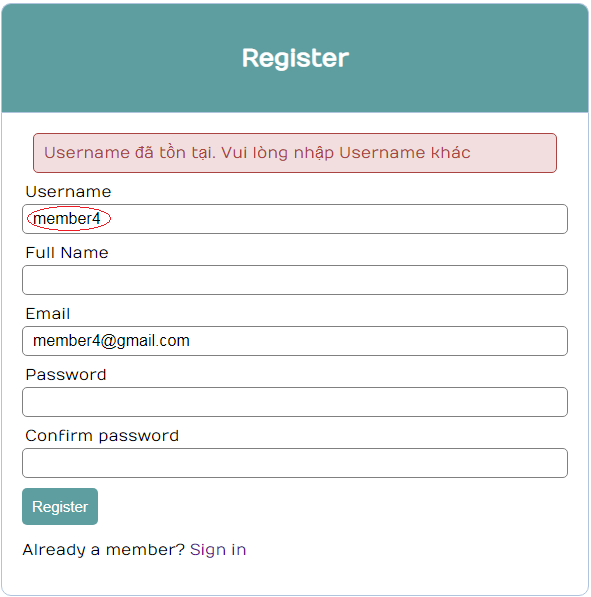
* Code sau khi đã kích hoạt tài khoản thành công:



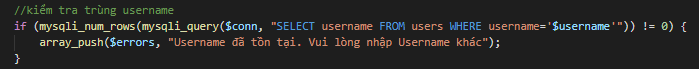
* Trường hợp khi đăng ký mà người dùng không điền thông tin vào các field mà lại nhấn trực tiếp “**Register**” thì hệ thống sẽ thông báo người dùng cần điền thông tin vào field.



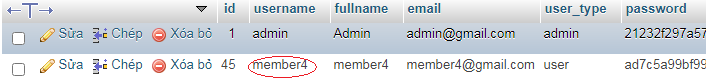
* Trường hợp đăng ký bị trùng username hệ thống: Sau khi nhấn nút button “**Register**” thì hệ thống sẽ thông báo username đã tồn tại vì trong database đã tồn tại tài khoản này trước đó.



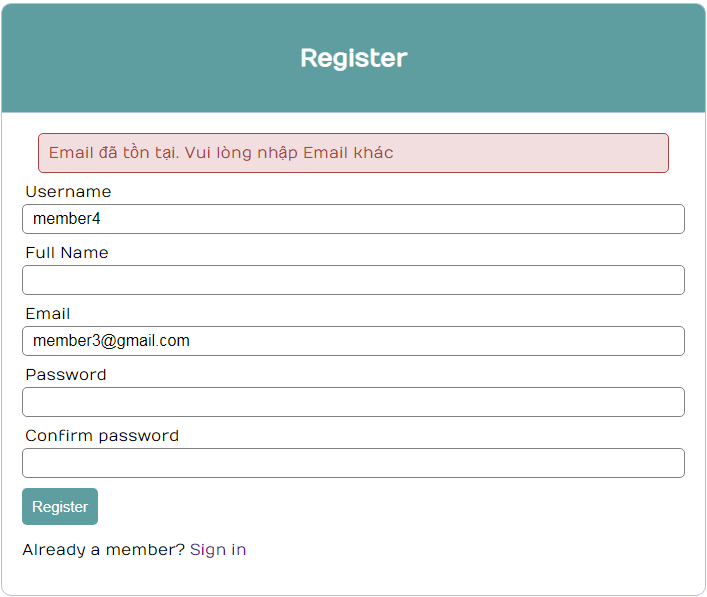
* Code kiểm tra username bị trùng nhau:

+ Hệ thống sẽ thực hiện kiểm tra trên database nếu bị trùng username thì hệ thống sẽ thông báo username đã tồn tại.

* Username đã tồn tại trong database:

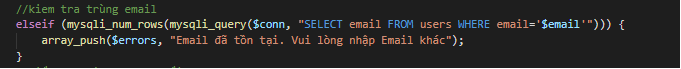


* Trường hợp khi đăng ký tài khoản mới nhưng dùng email đã tồn tại trước đó thì hệ thống sẽ thông báo: “Email đã tồn tại”

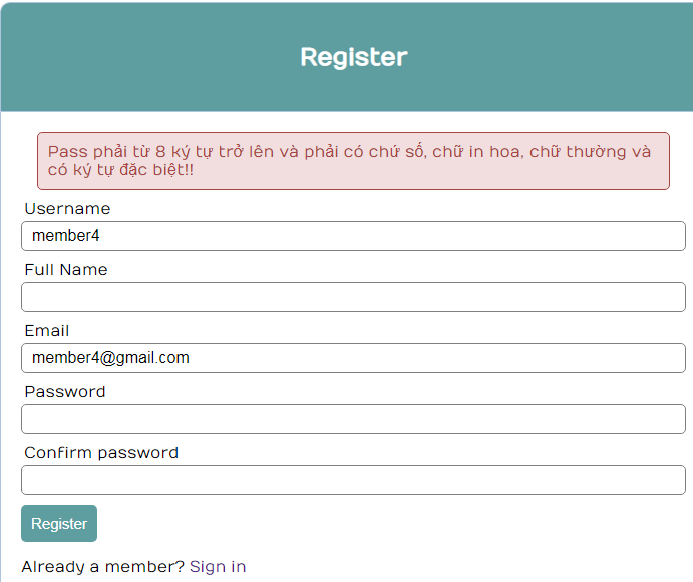


* Code thực hiện kiểm tra email đã tồn tại hay không:

+ Hệ thống sẽ thực hiện kiểm tra điều kiện nếu như email trên database bị trùng thì sẽ thực hiện đẩy ra thông báo “Email đã tồn tại. Vui lòng nhập Email khác”.

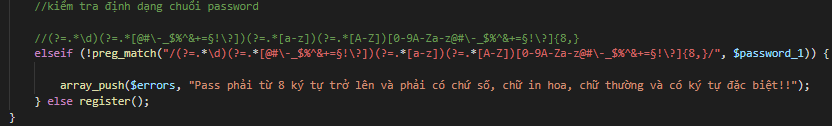


* Trường hợp kiểm tra password không đặt theo đúng yêu cầu hệ thống sẽ báo:



* Code kiểm tra định dạng password:

+ Hệ thống sẽ thực hiện xác minh chuỗi, kiểm tra xem password có thoả mãn yêu cầu: phải từ 8 ký tự trở lên và phải có chữ số, chữ in hoa, chữ thường và có kí tự đặc biệt hay không. Nếu khi đặt password không thỏa điều kiện như trên thì hệ thống sẽ báo lỗi.



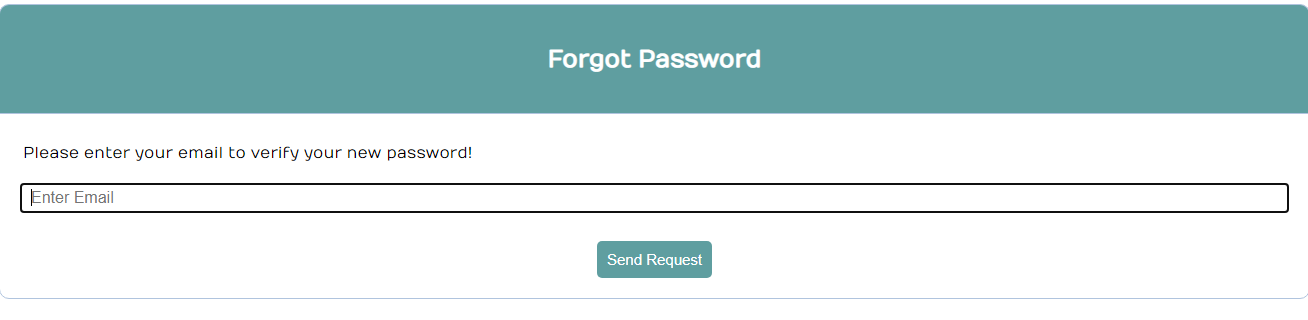
# Forgot password

## Tổng quan

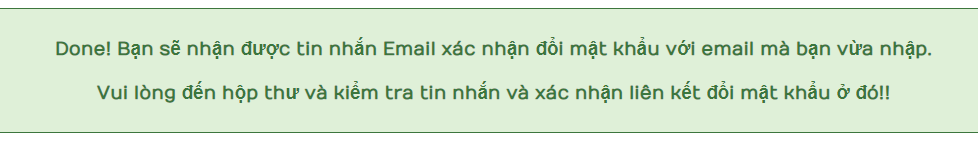
Khi người dùng quên mật khẩu tài khoản đăng nhập của họ, người dùng nhấn vào “**Forgot password?”**



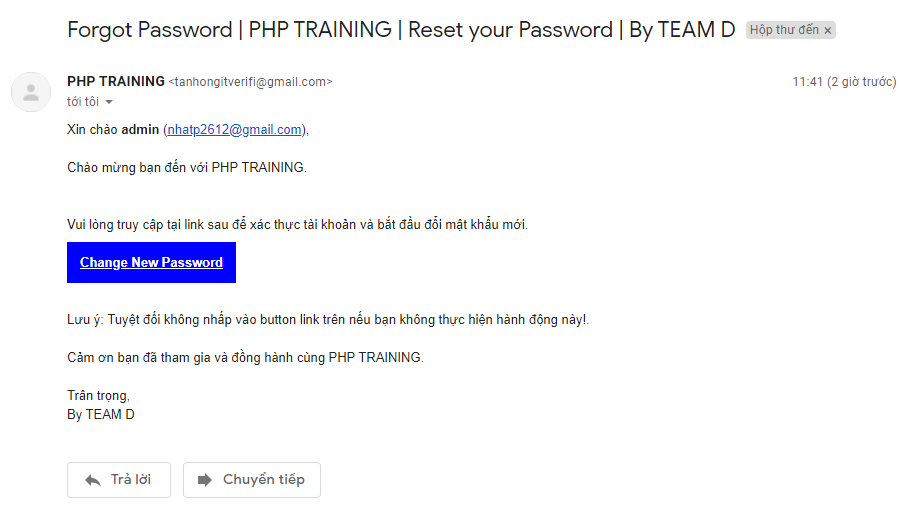
Sau khi nhấn sẽ chuyển đến form nhập email của họ:



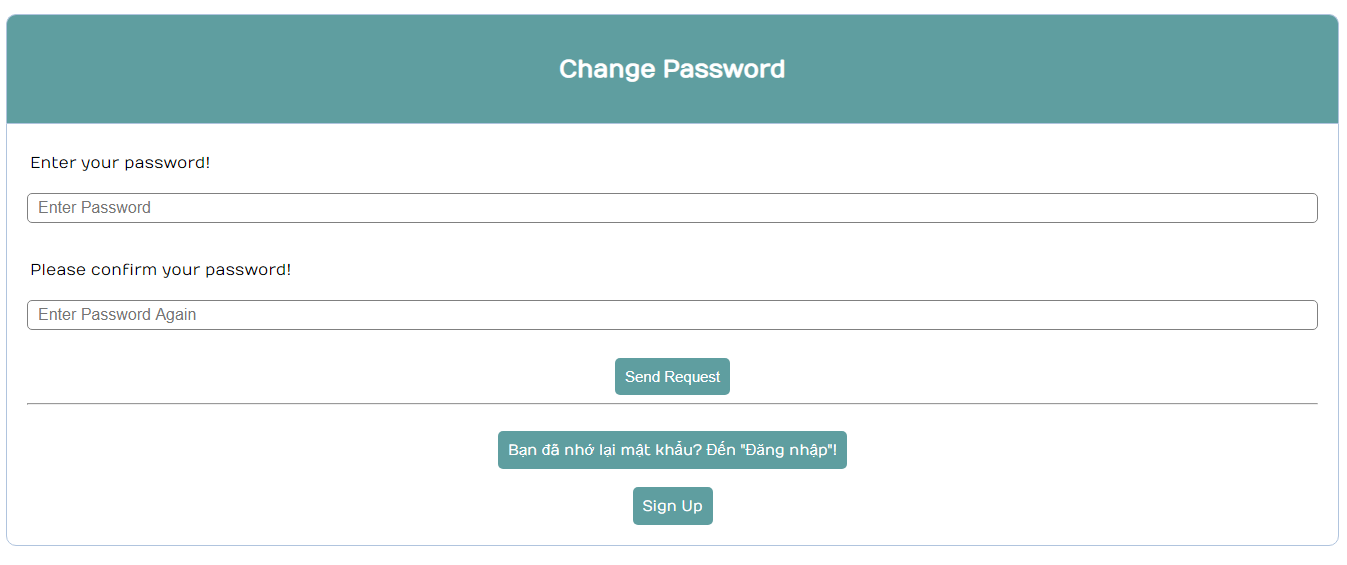
Sau khi nhấn kích hoạt với gmail đã có tồn tại trong database thì sẽ nhận được thông báo sau:



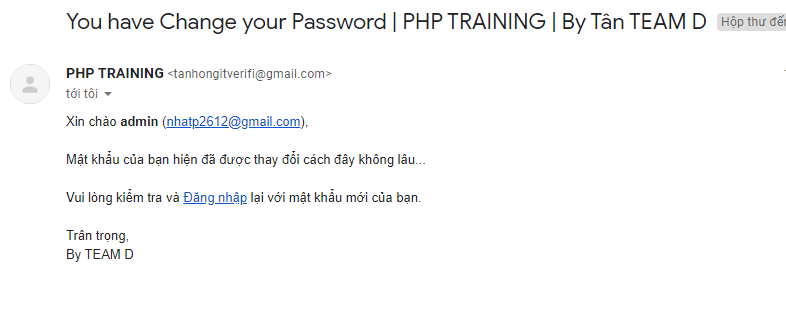
Ở form này yêu cầu người dùng phải nhập email để gửi đường dẫn xác thực kích hoạt lên Email và người dùng phải nhấn vào button kích hoạt trong email.



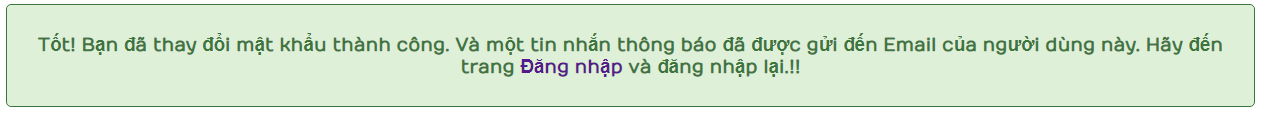
Sau khi người dùng truy cập email và click vào button “**Change new password**”. Sẽ chuyển đến form **change password.**



Nếu người dùng nhập đúng sẽ hiện thông báo thành công và sẽ gửi email thông báo đồng thời lưu mật khẩu đến dữ liệu người dùng đó trên database:

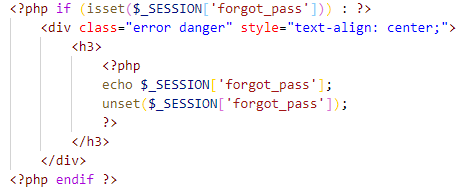


Trong thông báo có 1 đường dẫn đến trang login: “Đăng nhập” nếu người dùng click vào thì sẽ chuyển về trang login.

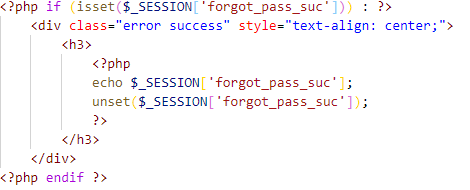


## Phân tích code

1. File forgot-password/index.php



* Nếu tồn tại session forgot\_pass thì hiển thị thông báo session forgot\_pass và gỡ giá trị trong session biến forgot\_pass



* Nếu tồn tại session forgot\_pass\_suc thì hiển thị thông báo session forgot\_pass\_suc và hủy session forgot\_pass\_suc

1. File forgot-password/request.php



Thêm thư viện PHPMailer và Exception



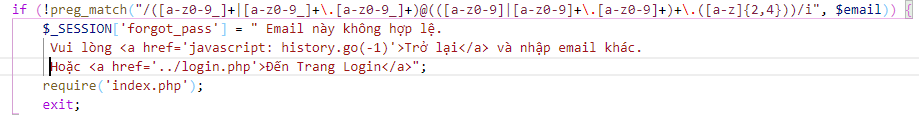
* Thêm file function.php và config.php



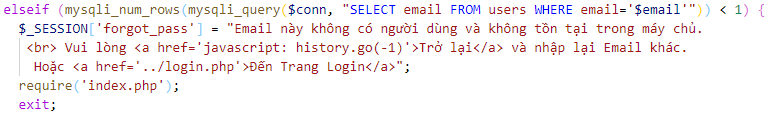
* Nếu không tồn tại post email thì trở về trang index.php



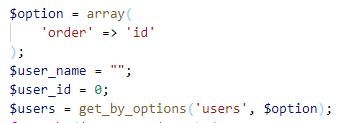
* Khai báo biến toàn cục $conn



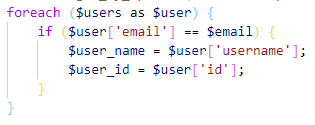
* Kiểm tra biến $email có phải là email hay không, nếu không thì gán thông báo cho $\_SESSION['forgot\_pass'], hiển thị trang index.php và thoát



* Nếu email không tồn tại trong database thì xuất thông báo, hiển thị trang index.php và thoát



* Tạo các biến cần sử dụng



* Sử dụng vòng lặp kiểm tra mảng $users, nếu email có tồn tại trong mảng $users thì gán $user\_name và $user\_id với $user tương ứng



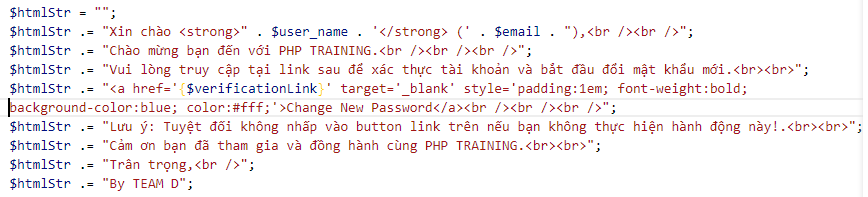
* thêm file autoload.php và setting.php



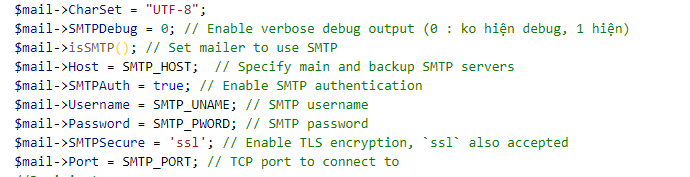
* Tạo 1 PHPMailer có giá trị == true



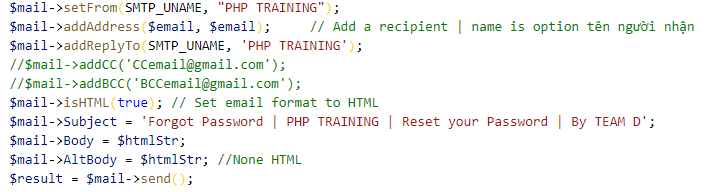
* Mã hóa user id bằng md5 và gán cho biến $verificationCode



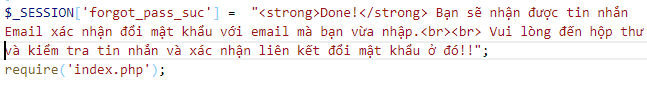
* Nội dung email được gửi



* Setting server



* Setting mail người được nhận

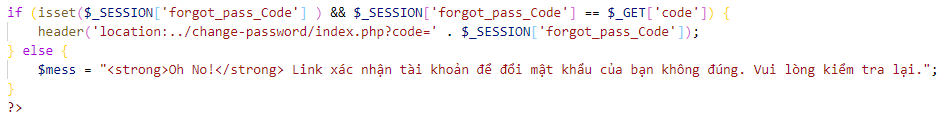


* Gán thông báo cho session và hiển thị trang index.php

1. File forgot-password/result.php



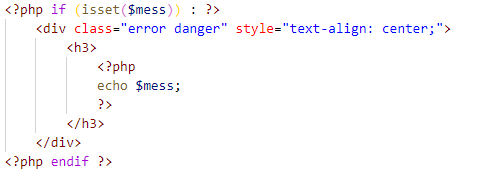
* Nếu không tồn tại get code hoặc get code không có dữ liệu thì trở về trang index.php



* Nếu trong session tồn tại forgot\_pass\_Code và forgot\_pass\_Code == get code thì chuyển hướng đến changepassword /index.php ? code=…



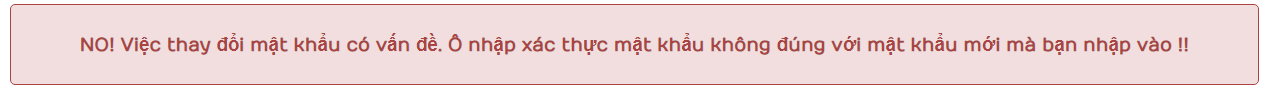
* Ngược lại thì gán thông báo lỗi cho biến $mess

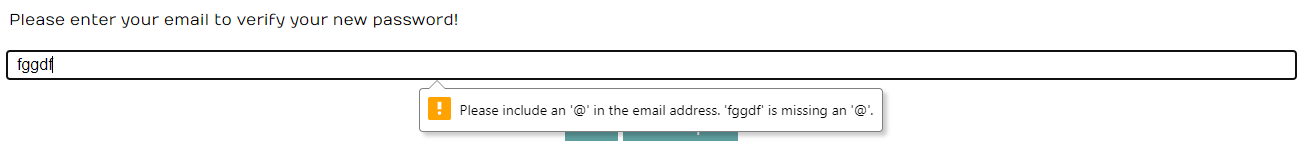


* Nếu tồn tại $mess thì hiển thị $mess

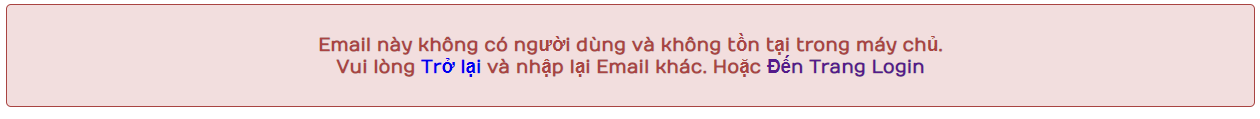
## Các trường hợp ngoại lệ

* Nếu người dùng confirm password mà không trùng với password mới thì sẽ báo lỗi và yêu cầu người dùng nhập lại.





* Không nhập đúng định dạng email

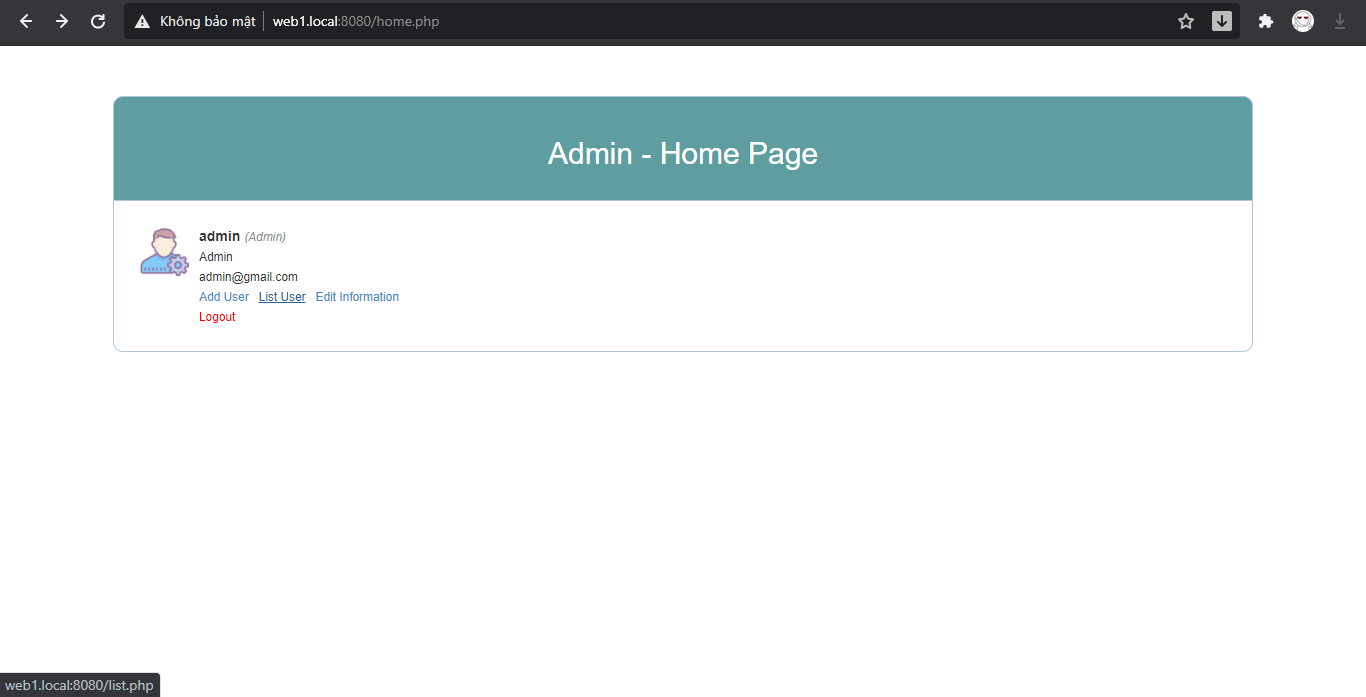


* Email không tồn tại

# View List User (Admin)

## Tổng quan

- Tại trang home.php, người dùng chọn "List User":

Click vào "List User".

- Sau đó người dùng sẽ được chuyển hướng tại trang list.php, là giao diện quản lý danh danh sách người dùng. Giao diện này cung cấp các chức năng sau:

+ Hiển thị danh sách người dùng.

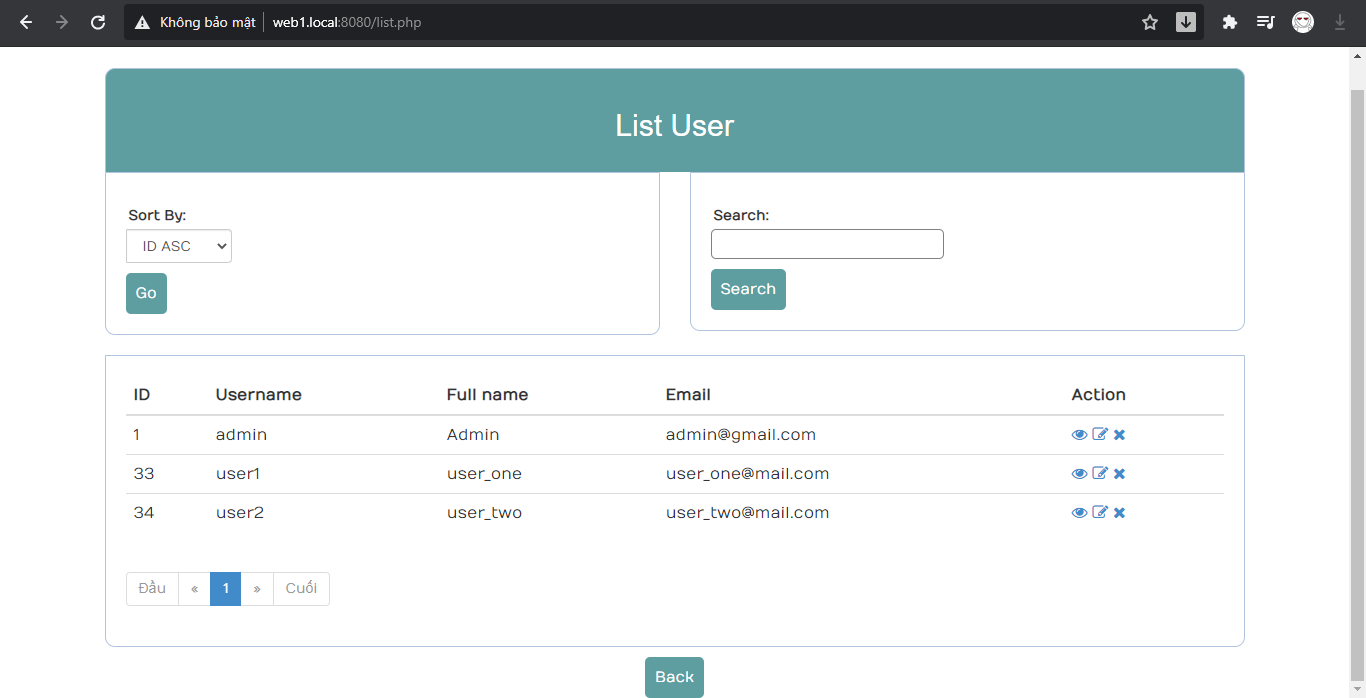
+ Sắp xếp danh sách người dùng.

+ Tìm kiếm người dùng trong danh sách.

+ Xem thông tin người dùng.

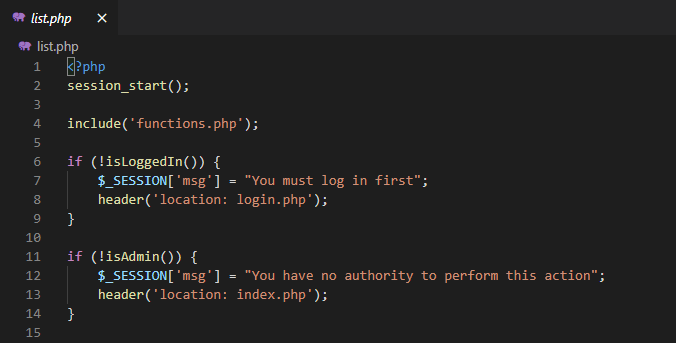
+ Sửa thông tin người dùng.

+ Xóa người dùng.

Giao diện quản lý danh sách người dùng.

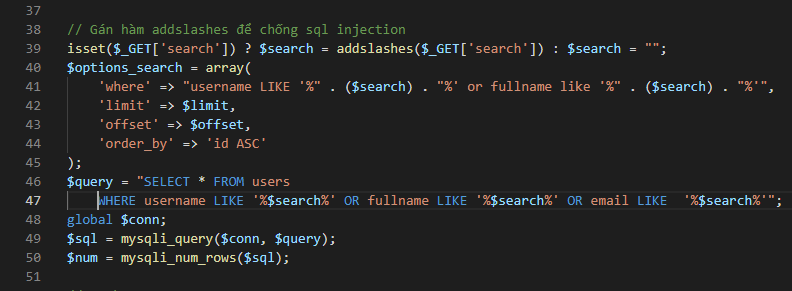
## Phân tích code

- Tại trang list.php, trước tiên phải kiểm tra 2 vấn đề: người dùng đã thực sự đăng nhập hay chưa, người dùng có phải là admin hay không. Nếu không thỏa 1 trong 2 điều kiện, thì chuyển hướng người dùng về trang khác.

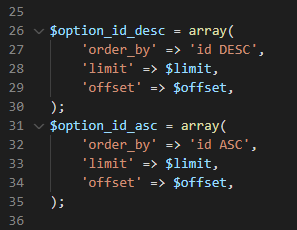


- Nếu thỏa các điều kiện trên, thì trang sẽ hiển thị toàn bộ danh sách người dùng.

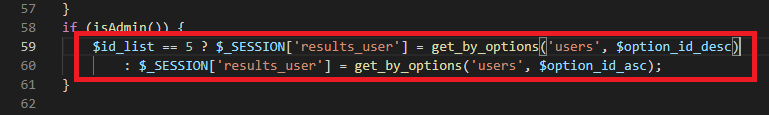
+ Đoạn code sau được sử dụng chung với chức năng Search. Chức năng này cho phép tìm kiếm gần đúng. Nếu ta không dùng chức năng Search, thì giá trị $search = "". Vậy sau khi thực thi truy vấn cơ sở dữ liệu, kết quả trả về sẽ là là toàn bộ danh sách người dùng.



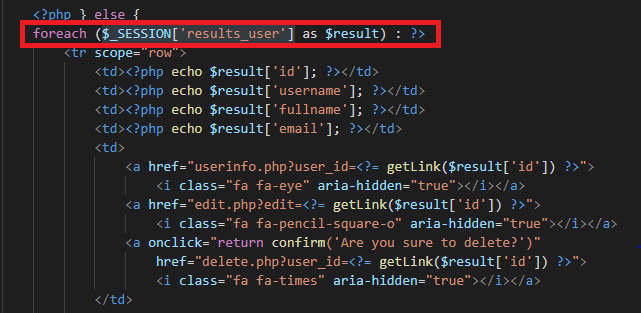
+ Mỗi biến $options\_search, $option\_id\_asc, $option\_id\_desc là một mảng lưu trữ các giá trị làm tham số cho câu truy vấn cơ sở dữ liệu. Để xác định thứ tự sắp xếp, giới hạn số lượng kết quả sẽ được hiển thị trên trang, ...



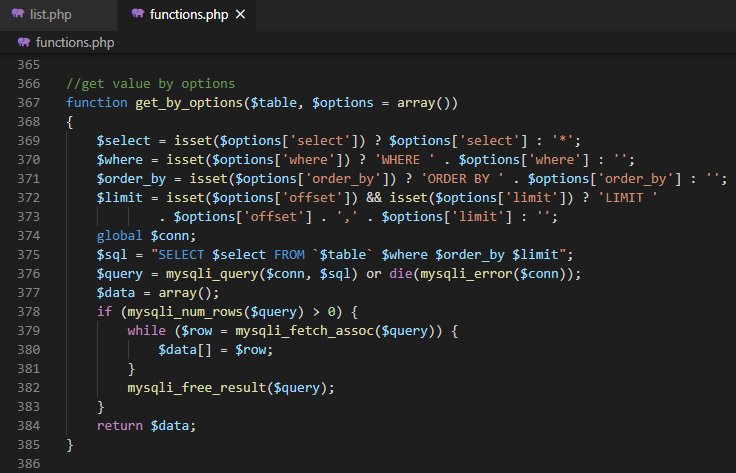
+ Trong đoạn code sau, hàm get\_by\_options() được dùng để lấy danh sách người dùng và lưu vào $\_SESSION['results\_user'].



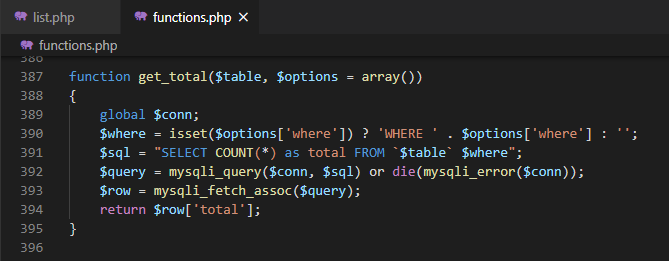
+ Duyệt mảng danh sách người dùng được lưu tại $\_SESSION['results\_user'] để lấy ra và hiển thị thông tin của từng người dùng:



+ Hàm get\_by\_options() được định nghĩa trong file functions.php. Tham số truyền vào (input) là tên bảng $table và $options là mảng lưu trữ các tham số cho câu truy vấn cơ sở dữ liệu (như là cho lệnh SELECT, WHERE). Hàm trả về (output) là một mảng danh sách người dùng mà thỏa điều kiện $options.



+ Hàm get\_total() được định nghĩa trong file functions.php. Tham số truyền vào (input) là tên bảng $table và $options là mảng lưu trữ các tham số cho câu truy vấn cơ sở dữ liệu (như là cho lệnh SELECT, WHERE). Hàm trả về (output) là tổng số lượng người dùng mà thỏa điều kiện $options. Kết quả trả về này được dùng để phục vụ cho việc phân trang.

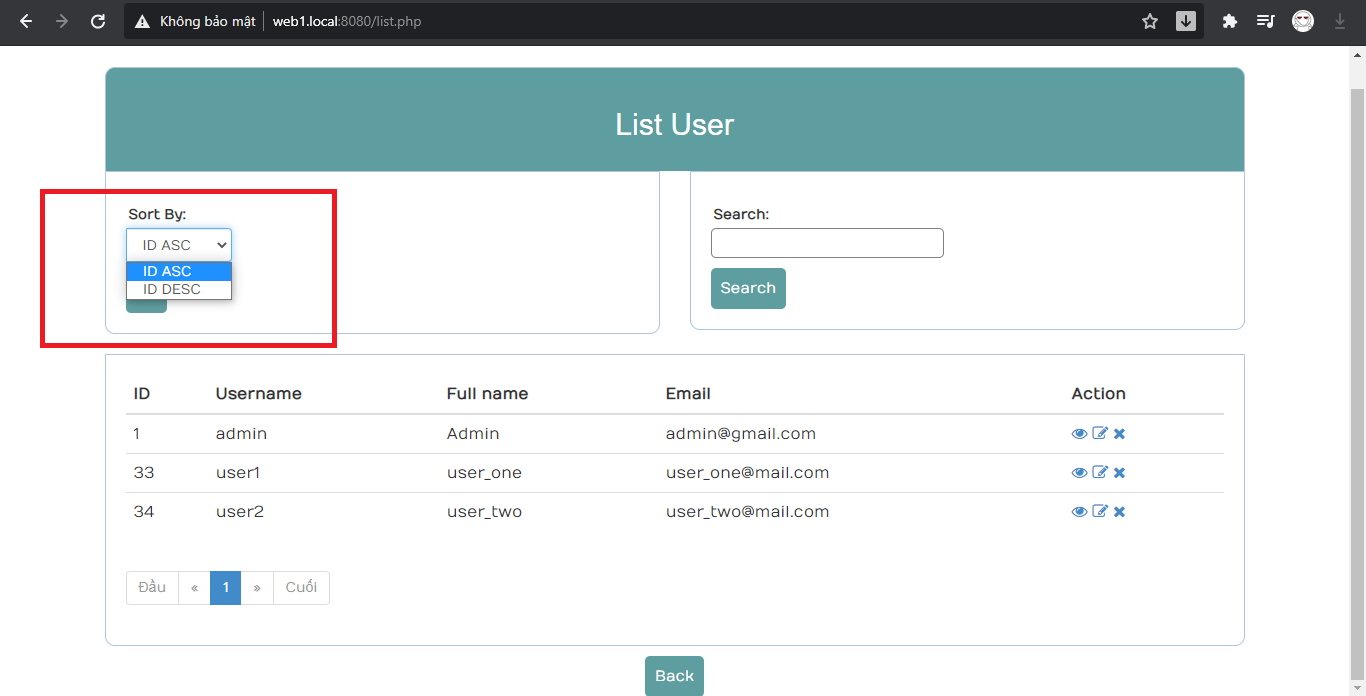


# Sort List (Admin)

## Tổng quan

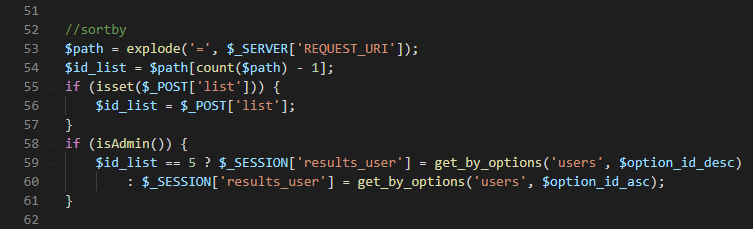
## 

- Tại giao diện quản lý danh sách người dùng, chức năng "Sort by" cung cấp 2 tùy chọn sắp xếp là tăng dần hoặc giảm dần theo giá trị ID. Ta chọn 1 trong 2 rồi nhấn nút "Go", bảng danh sách bên dưới sẽ được sắp xếp theo yêu cầu.

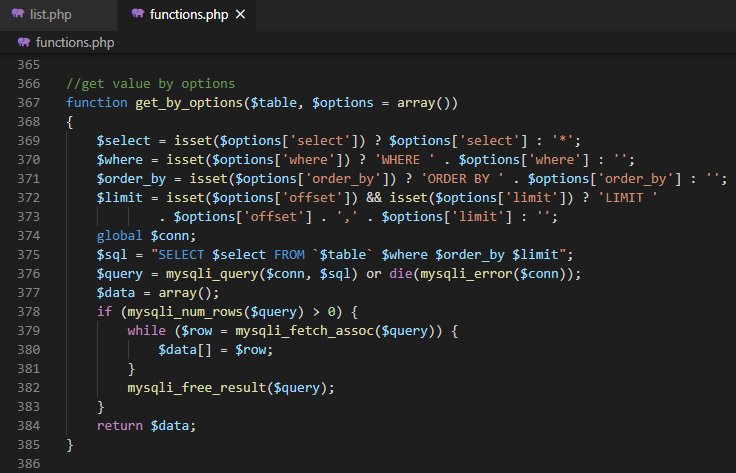
Giao diện quản lý danh sách người dùng - chức năng Sort.

## Phân tích code

- Mỗi biến $option\_id\_asc, $option\_id\_desc là một mảng lưu trữ các giá trị làm tham số cho câu truy vấn cơ sở dữ liệu. Trong đoạn code sau, hàm get\_by\_options() được dùng để lấy danh sách người dùng tùy theo trường hợp muốn sắp xếp tăng hoặc giảm dần theo ID.



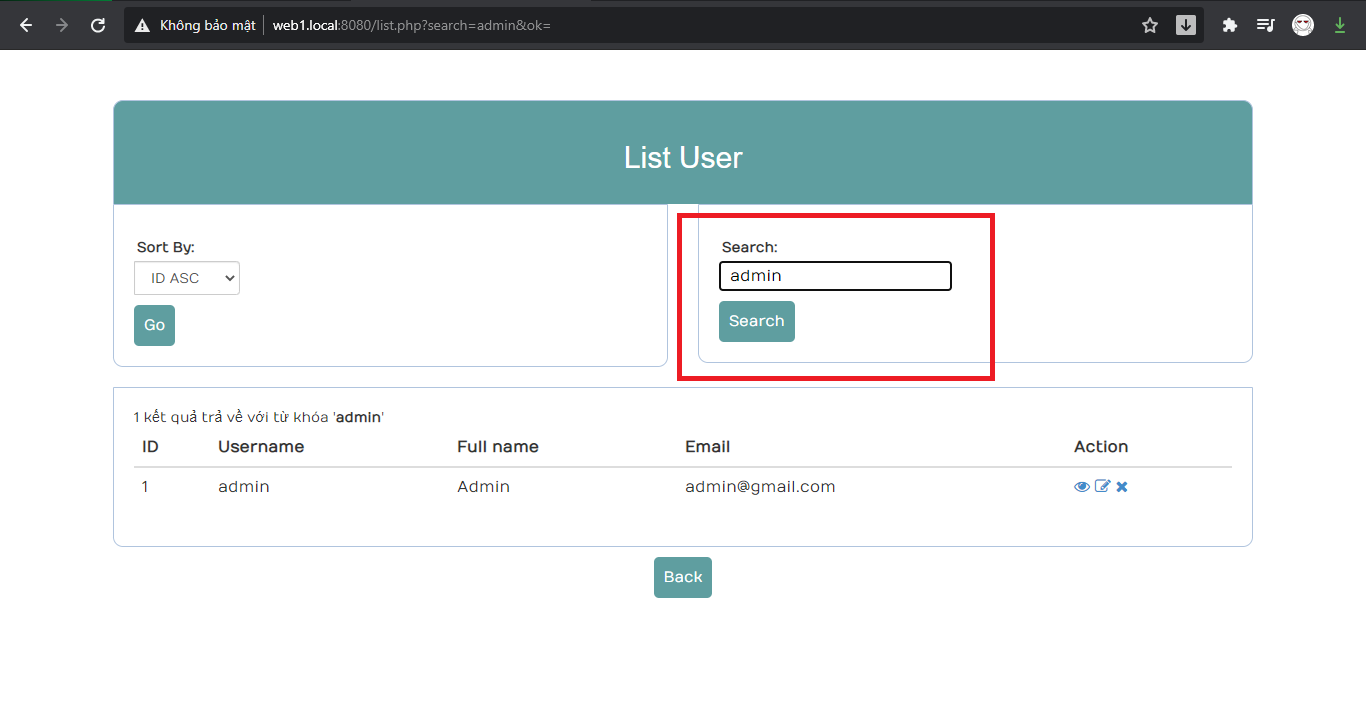
- Hàm get\_by\_options() được định nghĩa trong file functions.php. Tham số truyền vào (input) là tên bảng $table và $options là mảng lưu trữ các tham số cho câu truy vấn cơ sở dữ liệu (như là cho lệnh SELECT, WHERE). Hàm trả về (output) là một mảng danh sách người dùng mà thỏa điều kiện $options.



# Search User (Admin)

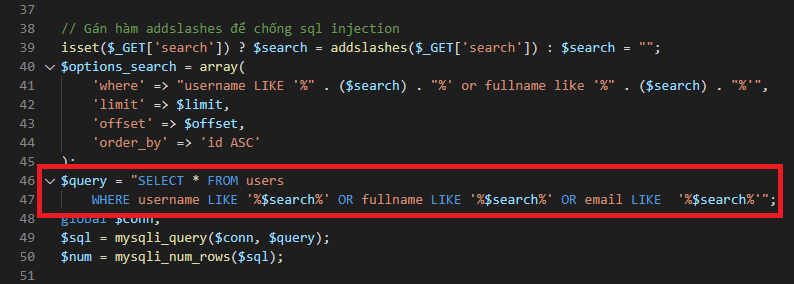
## Tổng quan

- Tại giao diện quản lý danh sách người dùng, chức năng "Search" cho phép tìm kiếm người dùng dựa theo 1 trong 3 tiêu chí: tìm theo Username, theo Full name hoặc theo Email. Ta nhập từ khóa cần tìm rồi nhấn nút "Search", bảng danh sách bên dưới sẽ hiển thị kết quả cần tìm.

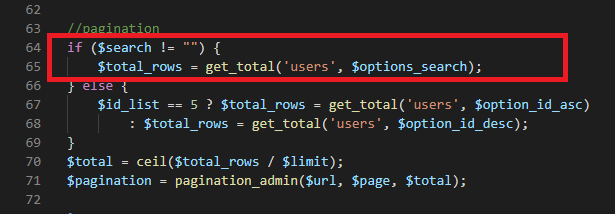
Giao diện quản lý danh sách người dùng - chức năng Search.

## Phân tích code

- Chức năng này cho phép tìm kiếm gần đúng. Nếu người dùng không dùng chức năng Search, thì giá trị $search = "". Vậy sau khi thực thi truy vấn cơ sở dữ liệu, kết quả trả về sẽ là là toàn bộ danh sách người dùng.



- Khi người dùng bắt đầu truy cập trang list.php, bảng danh sách người dùng sẽ được hiển thị nhờ có chức năng này.

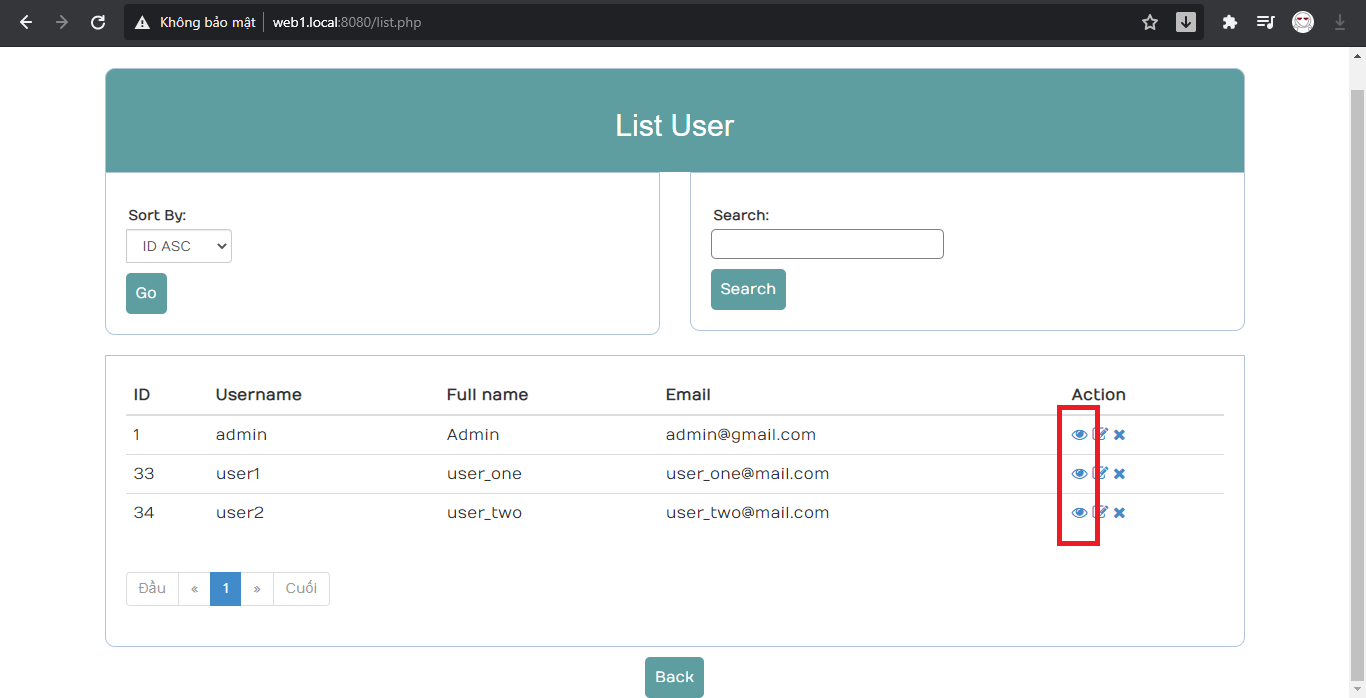


# View User Information (Admin)

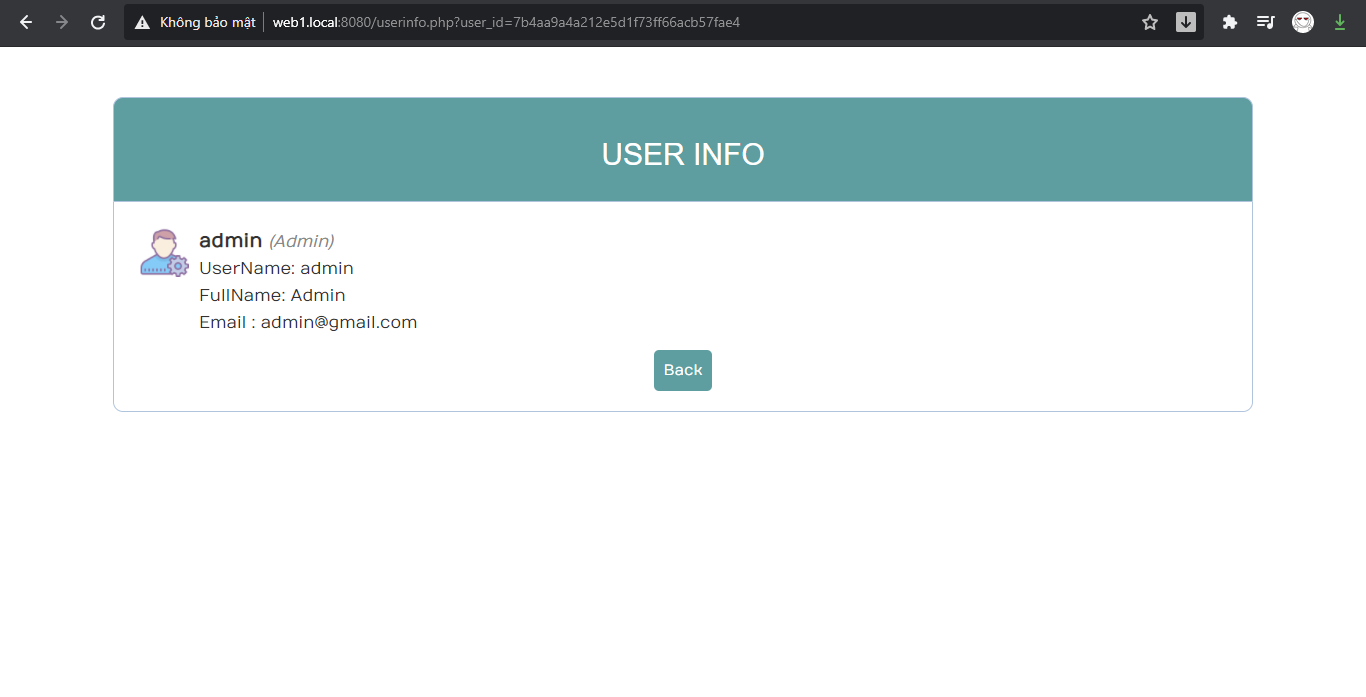
## Tổng quan

- Trong bảng danh sách người dùng, tại mỗi dòng thông tin đều cung cấp 3 chức năng: thêm, sửa, xóa dữ liệu của người dùng. Các chức năng đều được hiển thị bằng icon thuộc cột Action.

- Để xem chi tiết thông tin của một người dùng nào đó trong bảng, ta cần click vào icon thuộc dòng thông tin của người dùng đó:

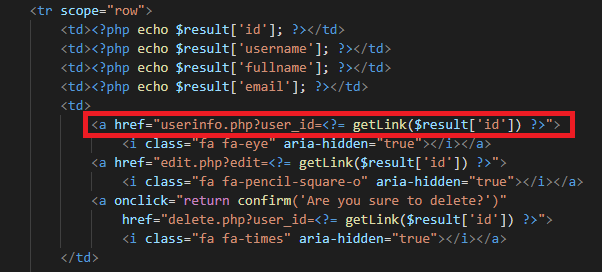


- Sau khi click vào icon xem thông tin, ta sẽ được chuyển hướng đến trang userinfo.php để xem thông tin chi tiết:

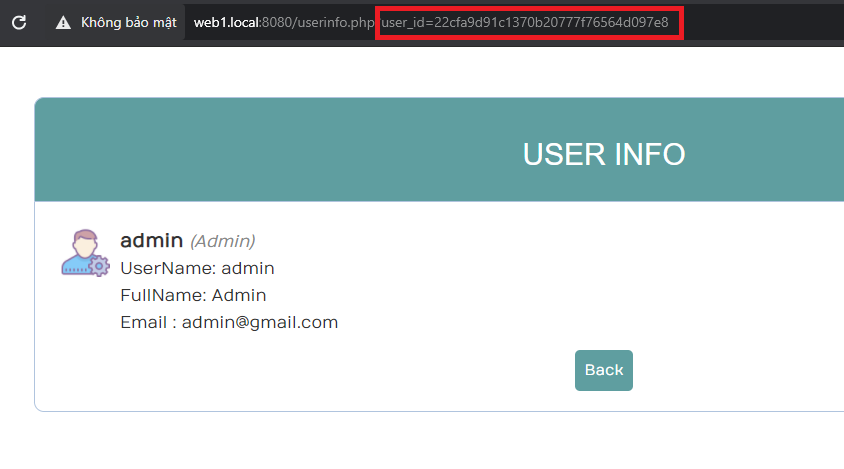


## Phân tích code

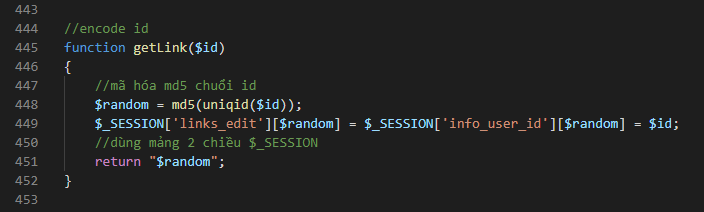
- Trong trang list.php, tại mỗi dòng thông tin người dùng trong danh sách sẽ có một nút xem thông tin chi tiết. Khi click vào nút này, ta sẽ được chuyển hướng tới trang userinfo.php với tham số là user\_id của người dùng đó.



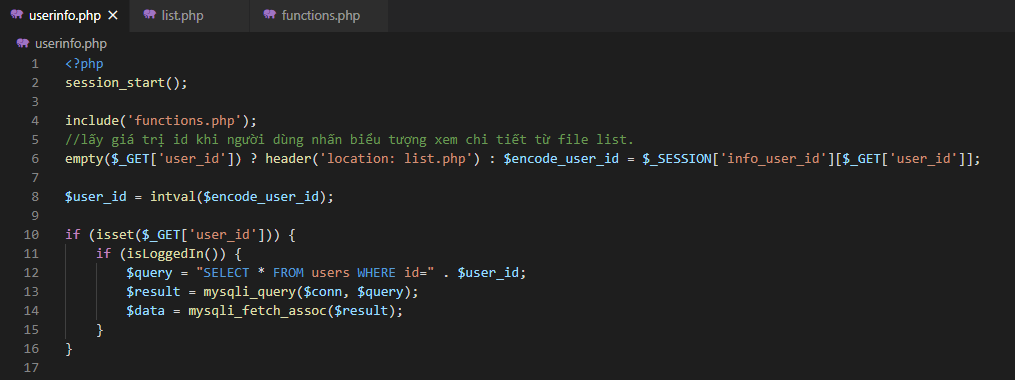
- Tham số là user\_id của người dùng sẽ được thay đổi bằng hàm getLink().



- Hàm getLink() được định nghĩa trong file functions.php. Tham số truyền vào (input) là một giá trị ID thật sự của người dùng trong CSDL. Trong hàm này, có sử dụng 2 hàm: md5() để mã hóa chuỗi theo MD5, uniqid() để tạo một ID duy nhất. Sau đó, tạo ra một $\_SESSION['links\_edit'][$random] để lưu giá trị ID. Hàm trả về (output) là một chuỗi $random được dùng để truy cập vào $\_SESSION['links\_edit'], chuỗi này ngăn người sử dụng biết được ID thật sự trong CSDL.



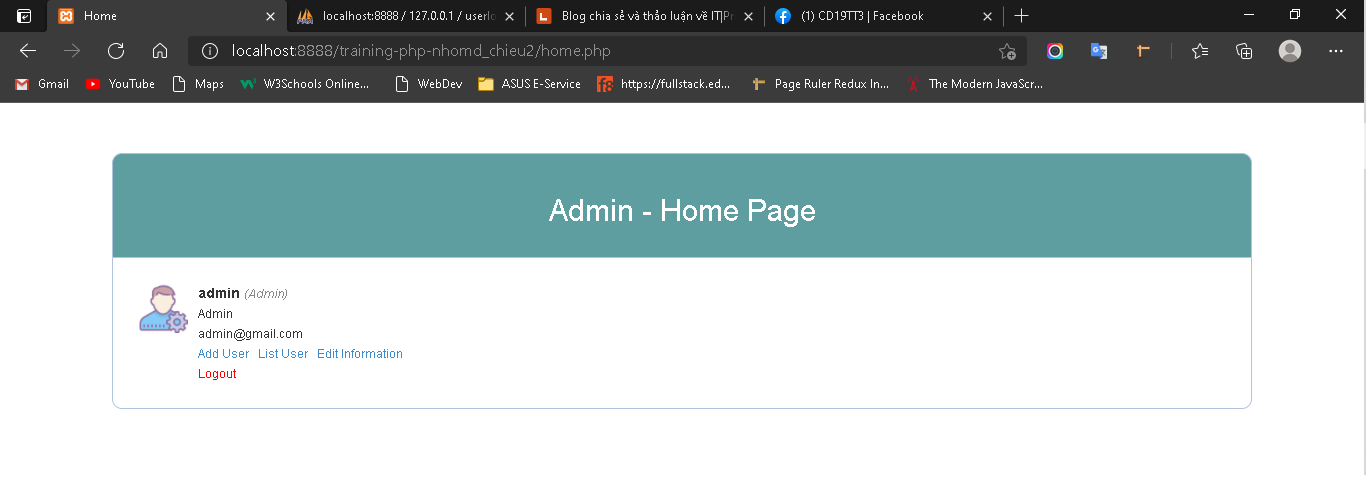
- Tại trang userinfo.php, đầu tiên kiểm tra xem có tham số user\_id được truyền vào hay không. Nếu không thì quay lại trang list.php (trang quản lý danh sách người dùng). Nếu có thì thông qua user\_id đó để lấy ra ID thật sự của người dùng trong cơ sở dữ liệu để thực hiện truy vấn thông tin.



# Add User (Admin)

## Tổng quan, phân tích code

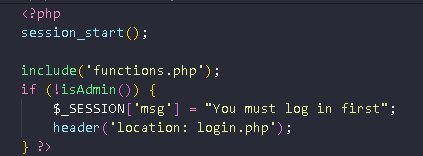
*Giao diện trang home.php*

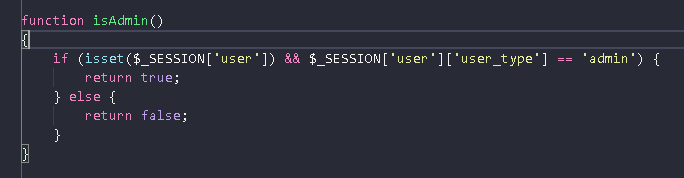


Chọn Add User -> Chuyển đến trang admin.php

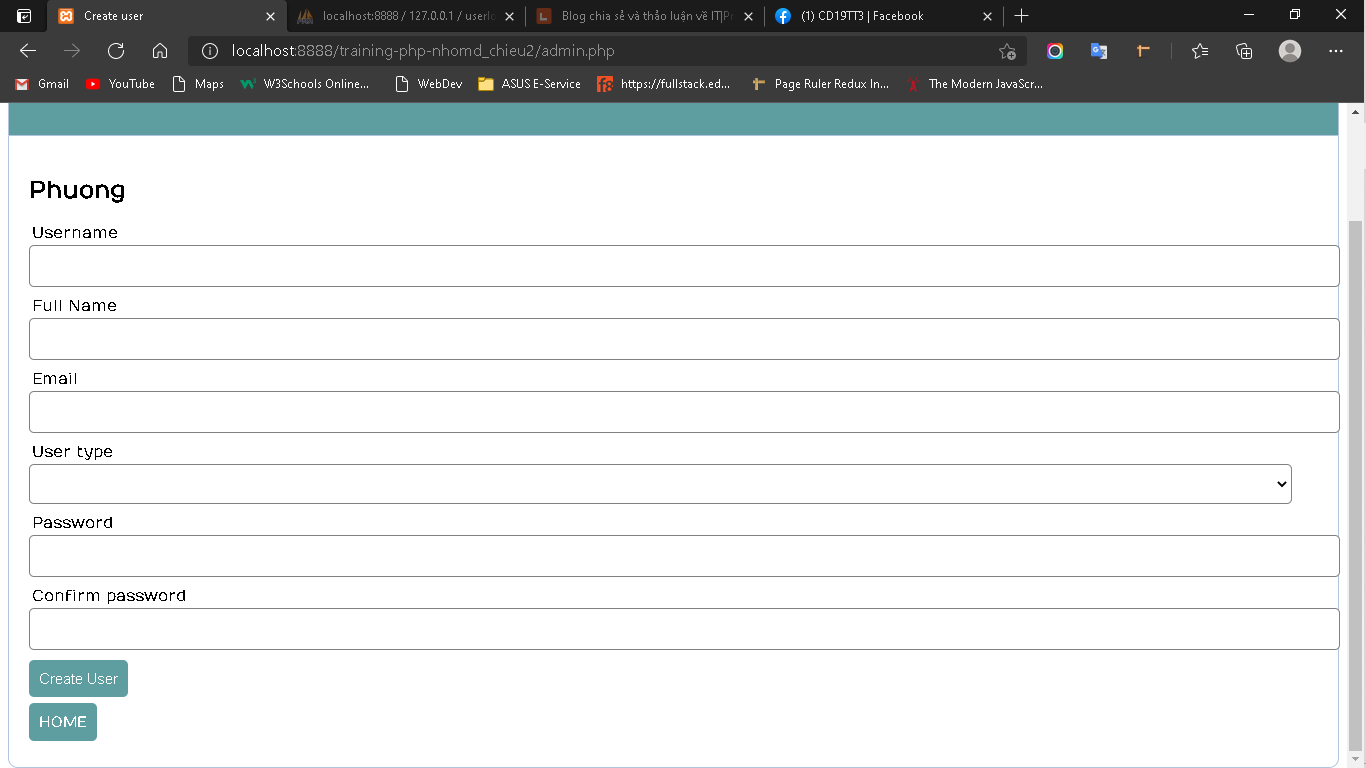


Kiểm tra xem đã Login và là admin thông qua session[‘user’] và session[‘user’][‘user-type’] là admin thông qua hàm isAdmin.





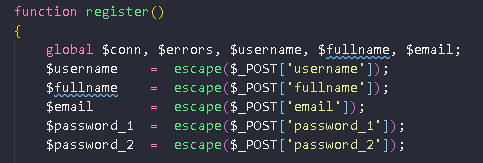
*Giao diện trang admin.php*



Ấn button HOME để quay lại trang home.php

Tiến hành nhập đầy đủ thông tin và ấn button Create User để tạo User.

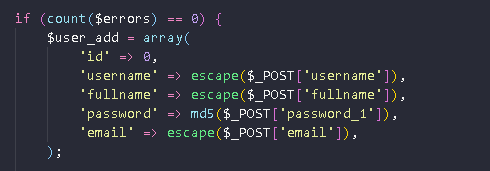
Sau khi nhập liệu hoàn chỉnh và ấn button Create User sẽ tiến hành xử lý code tạo user thông qua hàm register() (function.php).



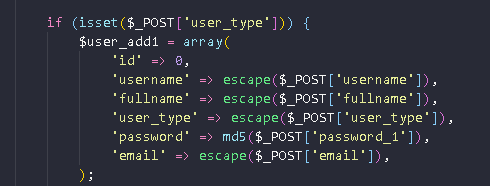
Tiến hành lấy tất cả các thông tin tại trang admin.php thông qua phương thức POST.



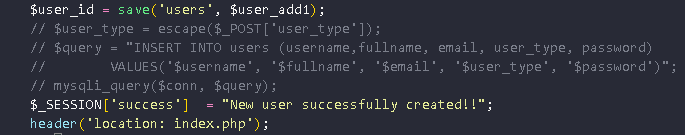
Tiến hành tạo User nếu mảng $errors 0 phần tử.



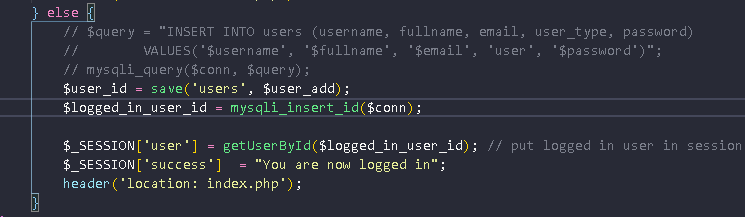
Tạo mảng $user\_add các thông tin cơ bản của user (không có user\_type).



Nếu $\_POST[‘user\_type’] tồn tại -> tạo mảng $user\_add1 với các thông tin cơ bản của User kèm user\_type.



Tiến hành insert mảng $user\_add1 vào cơ sở dữ liệu (save). Tạo Session[‘success’] và đi đến trang index.php

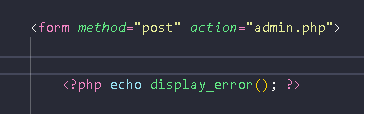


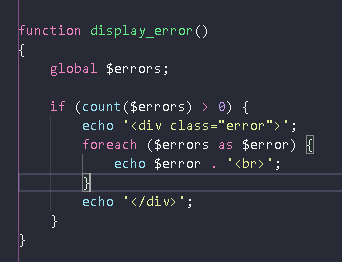
Nếu $\_POST[‘user\_type’] không tồn tại -> Tiến hành insert mảng $user\_add vào cơ sở dữ liệu với user\_type mặc định là ‘user’. Tạo Session[‘success’] và đi đến trang index.php.



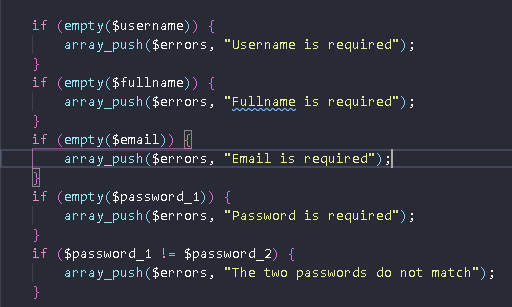
## Các trường hợp ngoại lệ

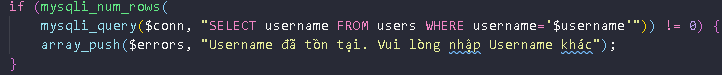
*Code xử lý:*

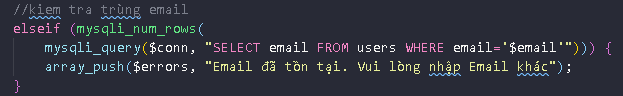


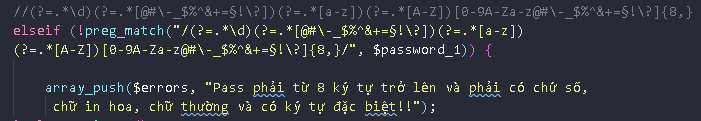


Hàm display\_error() (function.php) sẽ add 1 div class “error” vào giao diện admin.php khi tìm được lỗi khi nhập liệu thông qua mảng $errors



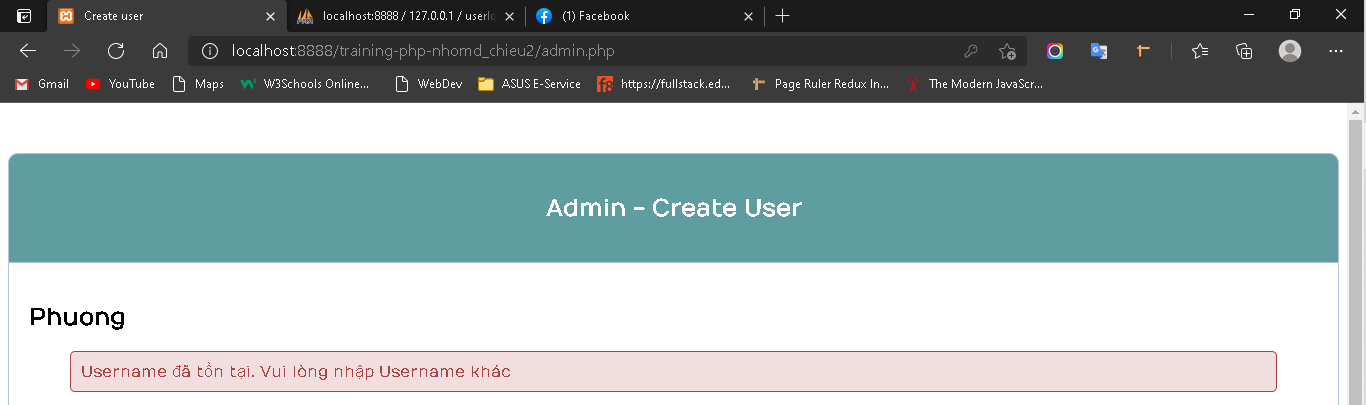
. 



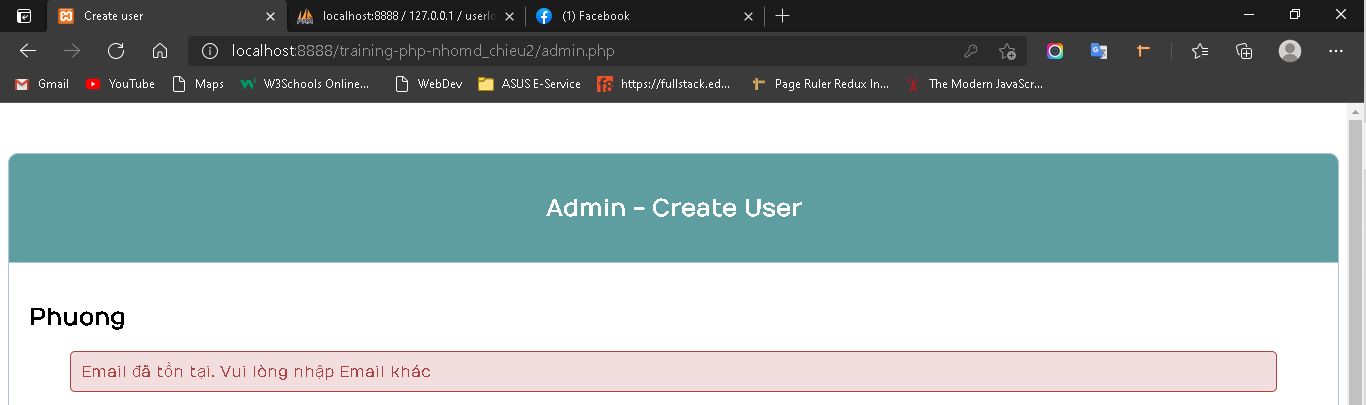


Các lỗi được tìm thấy khi nhập liệu sẽ được lưu vào mảng $errors.

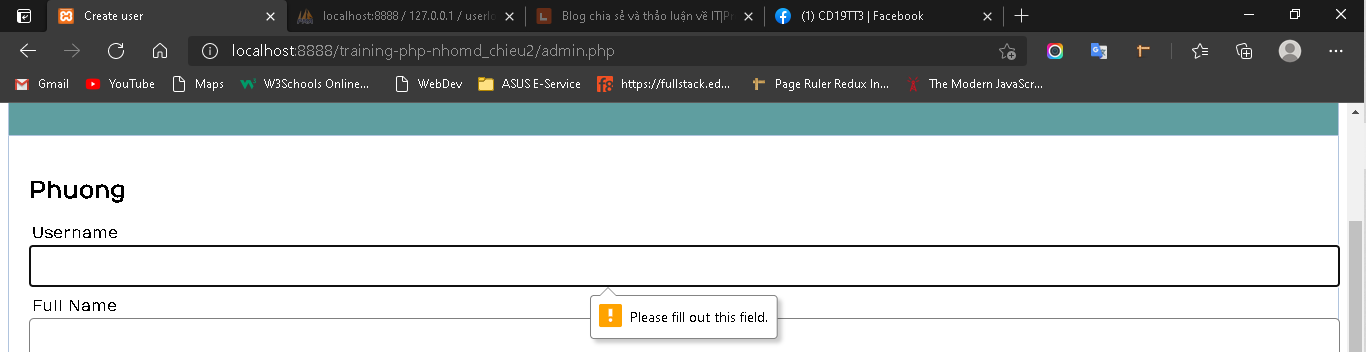
*Xử lý trên giao diện:*



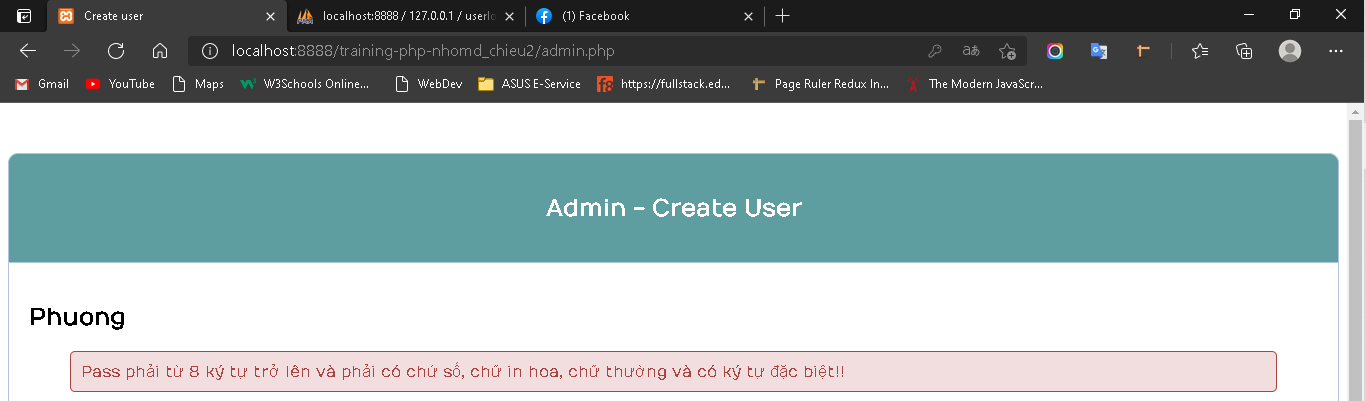
Nhập Username đã tồn tại sau khi ấn button Create User sẽ hiện thông báo nhắc nhở.



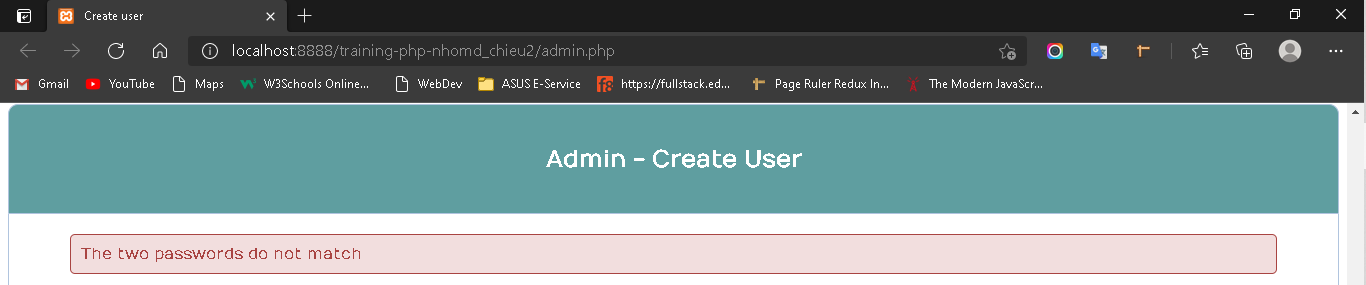
Nhập Email đã tồn tại sau khi ấn button Create User sẽ hiện thông báo nhắc nhở.



Nhâp trống sau khi ấn button Create User sẽ hiện thông báo nhắc nhở.



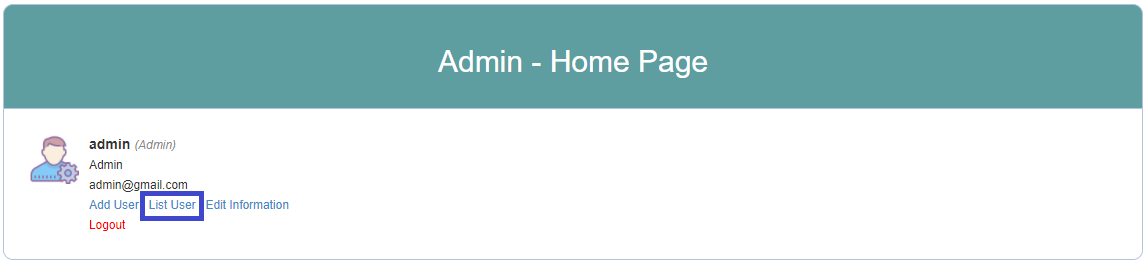
Nhập sai định dạng mật khẩu (Yêu cầu mật khẩu từ 8 kí tự trở lên, phải có chữ số, chữ in hoa, chữ thường và kí tự đặc biệt) sau khi ấn button sẽ hiện thông báo nhắc nhở.



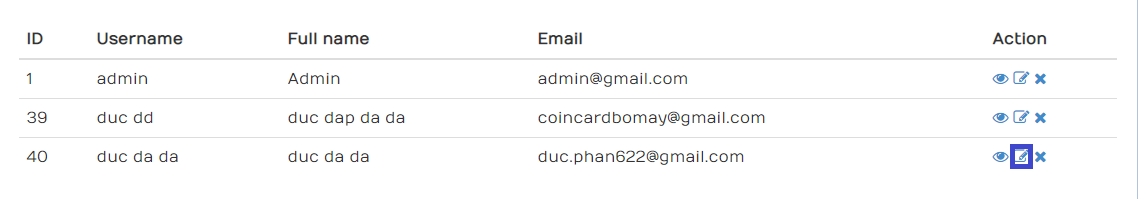
Nhập password và confirm password không trùng nhau sau khi ấn button Create User sẽ hiện thông báo nhắc nhở.

# Edit User (Admin)

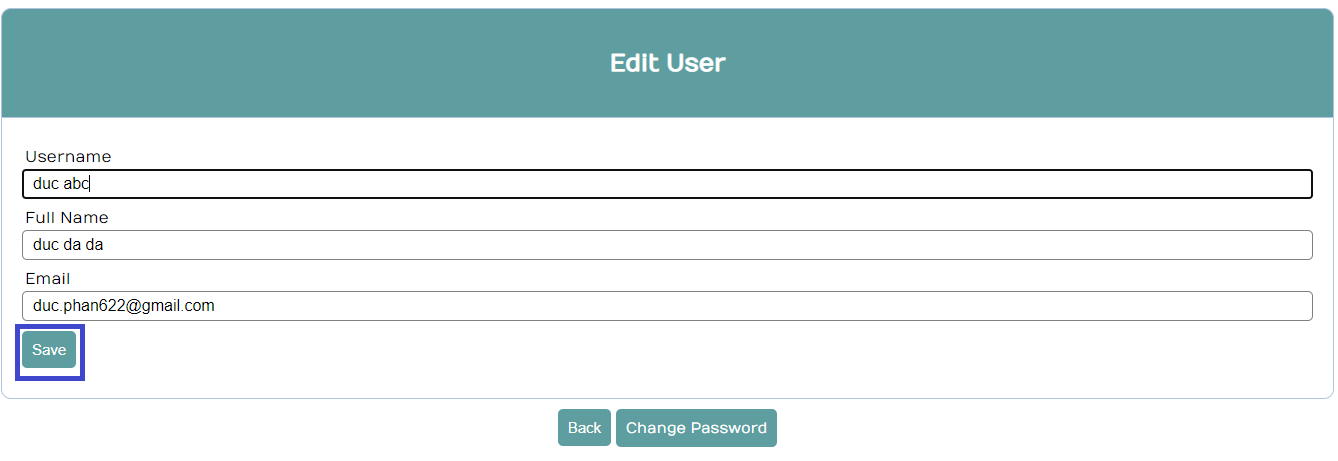
## Giao diện



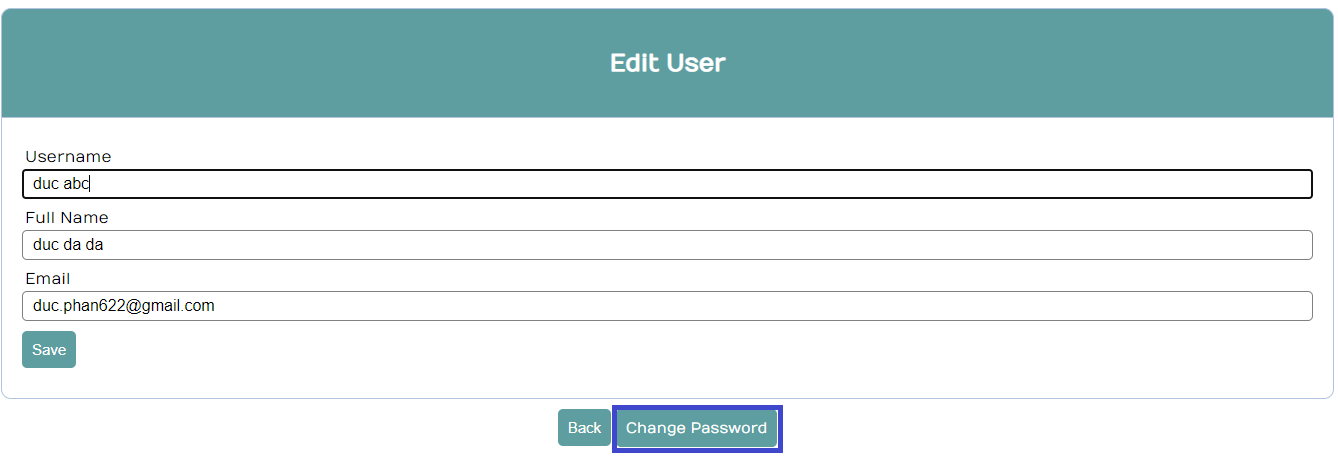
Đăng nhập bằng tài khoản có quyền admin sau đó click vào nút **List User**



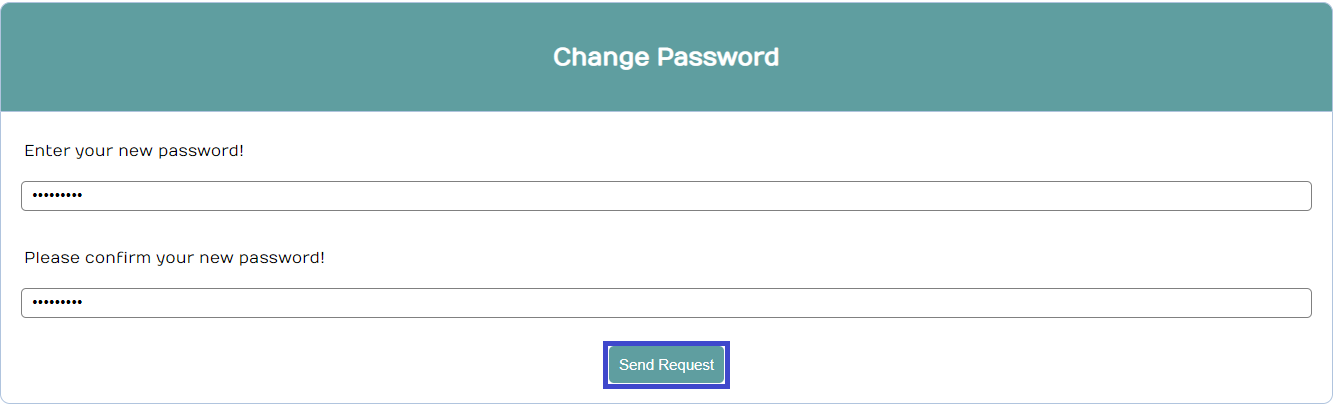
Click vào nút có biểu tượng hình **cây bút chì** của user bạn muốn chỉnh sửa, ở đây tôi chọn user có id 40



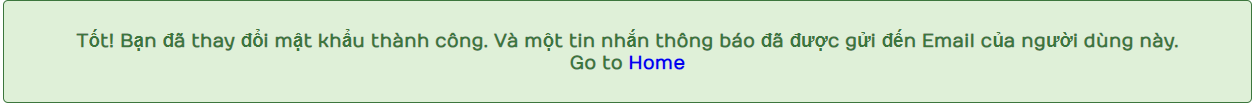
Nhập thông tin bạn muốn chỉnh sửa, sau đó nhấn nút **Save**



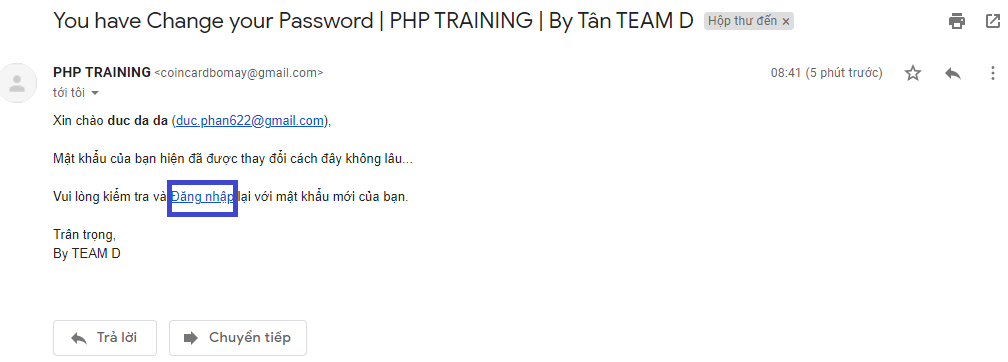
Trong trường hợp bạn muốn sửa password của user, click vào **Change Password**



Nhập password và confirm password sau đó nhấn **Send Request**



Sau khi thay đổi password thành công thì sẽ xuất hiện thông báo cho admin và một thông báo sẽ được gửi tới email của user



Trong thông báo của user, user click vào **Đăng nhập** để đăng nhập lại với mật khẩu mới

## Phân tích code

1. File change-password/admin-change.php



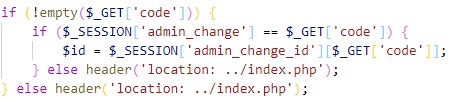
* Khởi tạo session



* Thêm file functions.php



* Nếu chưa được login, không phải admin và $\_GET[‘code’] rỗng thì chuyển hướng đến trang index.php



* Nếu $\_GET[‘code’] không rỗng và $\_SESSION[‘admin\_chane’]==$\_GET[‘code’] thì $id được gán bằng $\_SESSION[‘admin\_change\_id’][$\_GET[‘code’]], ngược lại thì chuyển hướng đến trang index.php

1. File change-password/request-admin.php

Capture

* Khởi tạo session

Capture

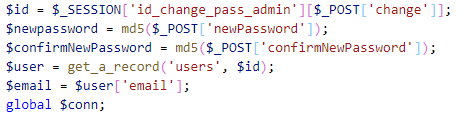
* Thêm file functions.php



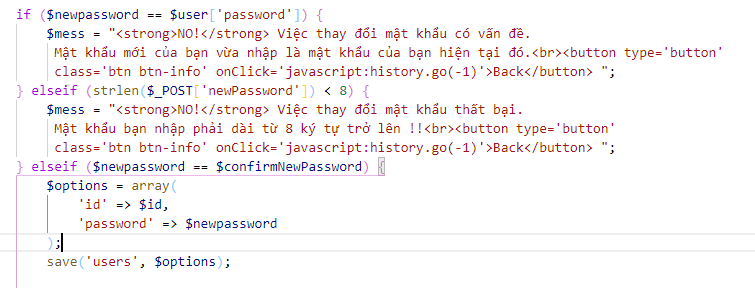
* Sử dụng thư viện PHPMailer



* Nếu $\_POST[‘change’] rỗng



* Gán $\_SESSION['id\_change\_pass\_admin'][$\_POST['change']] cho biến $id.
* Mã hóa md5 $\_POST[‘newPassword’] và gán cho biến $newPassword
* Mã hóa md5 $\_POST[‘confirmPassword’] và gán cho biến $confirmPassword
* Lấy thông tin user có $id
* Gán $user[‘email’] cho biến $email
* Tạo biến toàn cục $conn



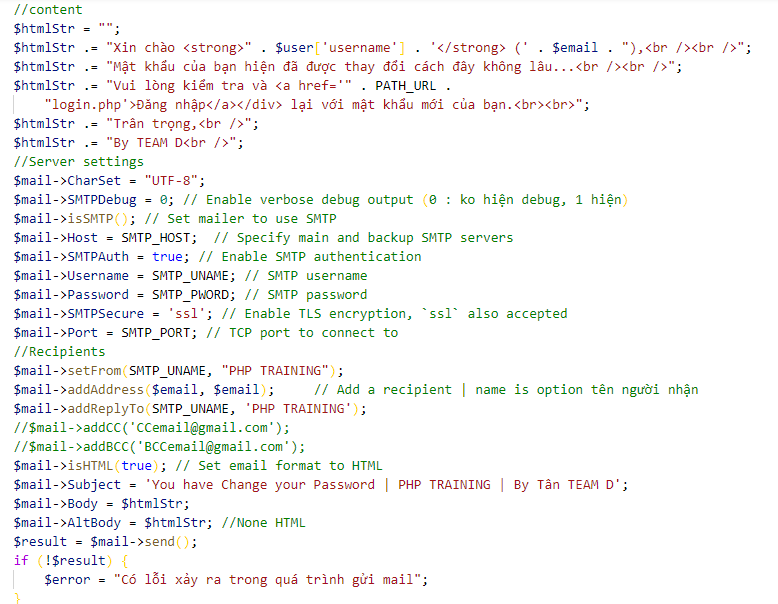
* Kiểm tra nếu $newPassword == $user[‘password’] thì gán thông báo cho biến $mess, tương tự với trường hợp chiều dài $\_POST[‘newPassword’] < 8.
* Trường hợp $newPassword==$confirmPassword thì tạo mảng có 2 phần tử id, password chứa giá trị $id và $newPassword, sau đó gán cho biến $options
* Lưu lại user với giá trị là $options



* Thêm file autoload.php và file setting.php



* Tạo $mail bằng PHPMailer



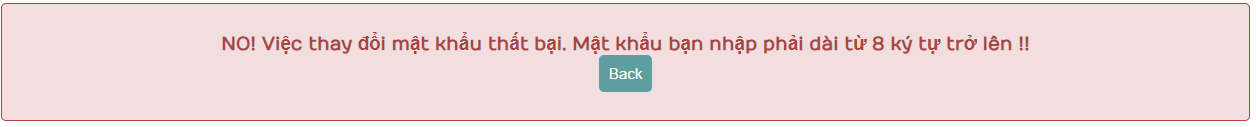
* Tạo content
* Setting server
* Setting email
* Nếu $result==true thì gán thông báo lỗi cho biến $error

1. File change-password/result.php

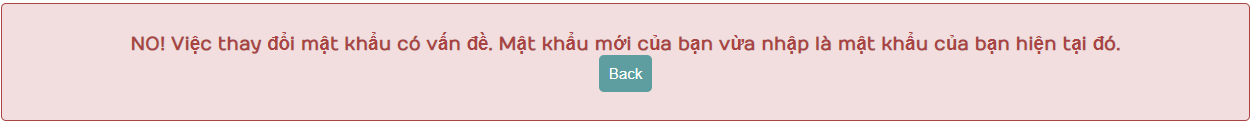


* Nếu tồn tại biến $mess thì hiển thị biến $mess
* Nếu tồn tại biến $mess\_success thì hiển thị biến $mess\_success

## Các trường hợp ngoại lệ



* Trường hợp admin nhập password có chiều dài dưới 8 ký tự thì sẽ có thông báo yêu cầu admin nhập password phải dài từ 8 ký tự



* Trường hợp admin nhập password cũ thì sẽ có thông báo yêu cầu admin nhập password mới



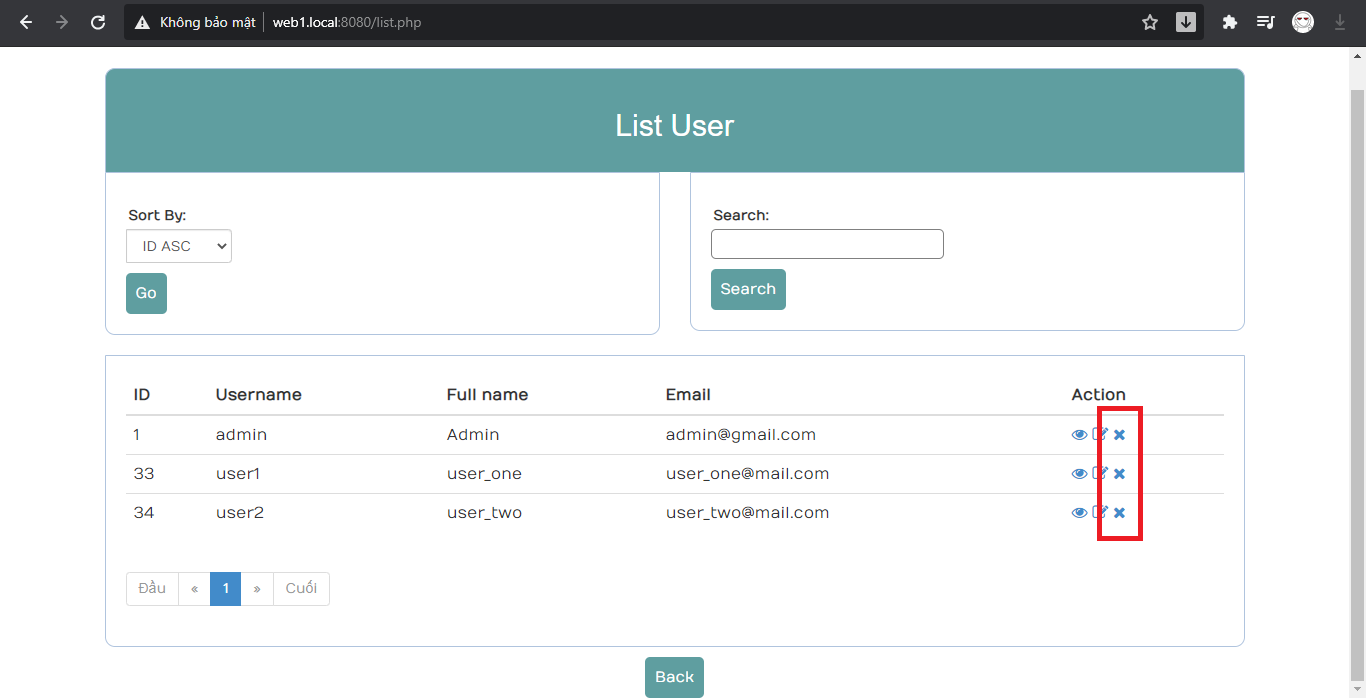
* Trường hợp password không giống confirm password thì bắt buộc phải nhập lại

# Delete User (Admin)

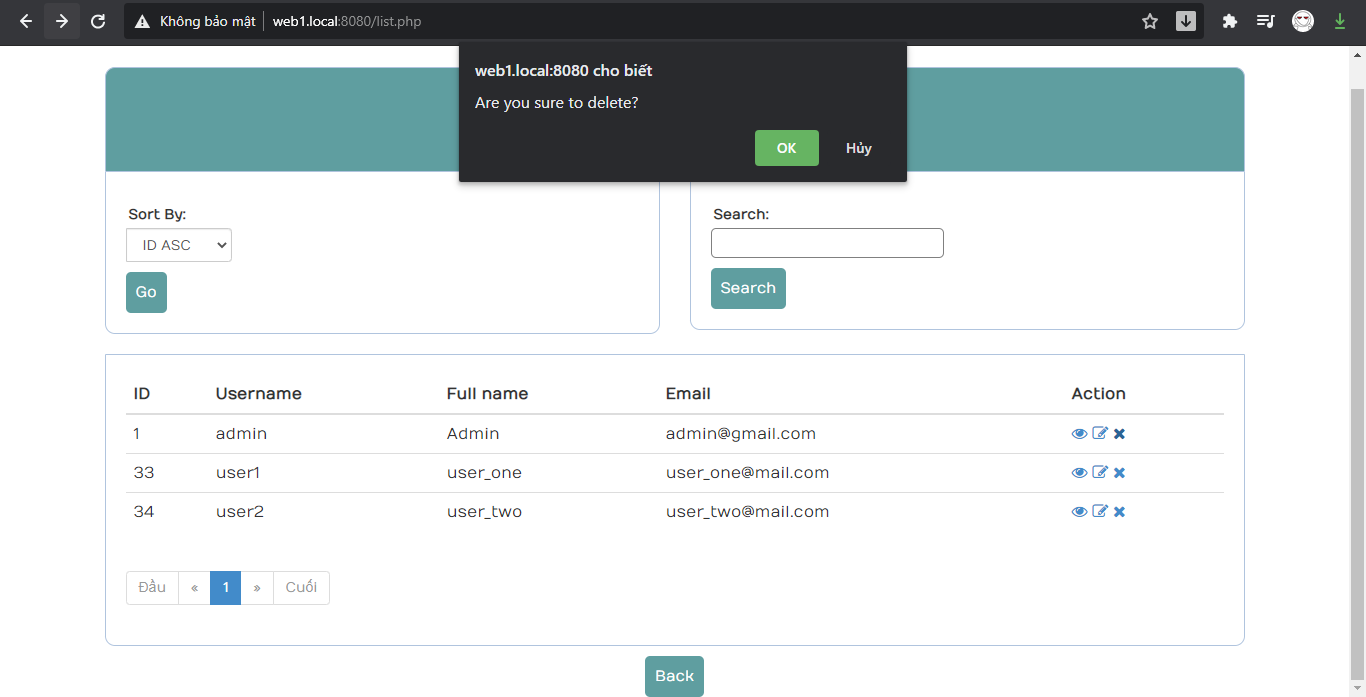
## Tổng quan

- Trong bảng danh sách người dùng, tại mỗi dòng thông tin đều cung cấp 3 chức năng: thêm, sửa, xóa dữ liệu của người dùng. Các chức năng đều được hiển thị bằng icon thuộc cột Action.

- Để xóa dữ liệu của một người dùng nào đó trong bảng, ta cần click vào icon hình chữ X thuộc dòng thông tin của người dùng đó:



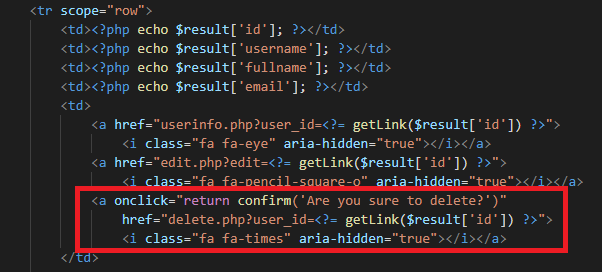
- Sau khi click vào icon xóa dữ liệu người dùng, một Message Block được hiển thị để xác nhận xem ta có thật sự muốn xóa hay không. Nếu có thì sẽ chuyển tới trang delete.php để thực hiện xóa dữ liệu, sau khi xóa xong sẽ quay lại trang list.php.



## Phân tích code

- Trong trang list.php, tại mỗi dòng thông tin người dùng trong danh sách sẽ có một nút để xóa dữ liệu. Khi click vào nút này, ta sẽ được chuyển hướng tới trang delete.php với tham số là user\_id của người dùng đó.

- Tham số là user\_id của người dùng sẽ được thay đổi bằng hàm getLink(), nhằm ngăn người sử dụng biết được giá trị ID đó.



- Tại trang delete.php, đầu tiên kiểm tra xem có tham số user\_id được truyền vào hay không. Nếu không thì quay lại trang list.php (trang quản lý danh sách người dùng). Nếu có thì thông qua user\_id đó để lấy ra ID thật sự của người dùng trong cơ sở dữ liệu để thực hiện việc xóa. Sau đó, quay lại trang list.php.

