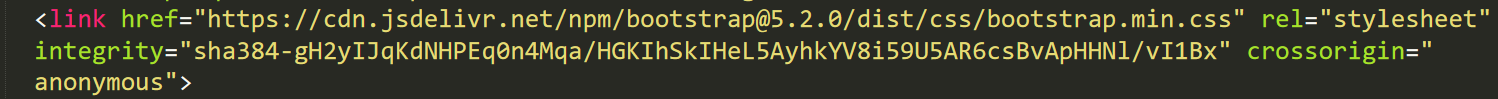
**Link source code:** <https://drive.google.com/drive/folders/1KgKzmWSRU7ZEvaHth3HYwNsFWbFBx8c0?usp=sharing>

**Github link:**

[Gia2912/Lession2: PHP Intern - Chung Khánh Gia - 04/08/2022 - Lampart Anwser Sheet (github.com)](https://github.com/Gia2912/Lession2)

1. **Liên kết**



Hình 1: Đường link dẫn đến Bootstrap

Sử dụng đường link được ở “*getbootstap.com*”, ta sẽ không cần phải viết thêm CSS vì đường link trên liên kết đến Bootstrap, nơi có sẳn CSS.

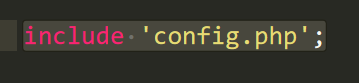
Ảnh có chứa văn bản

Mô tả được tạo tự động

Hình 2 : File config.php

File *config.php* sẽ là một file chính dùng để liên kết với MySQL.

Khi đó để sử dụng được file config.php thì sử dụng câu lệnh *include 'config.php';*



Hình 3:

1. **Database**

**Ảnh có chứa văn bản

Mô tả được tạo tự động**

Hình 4:

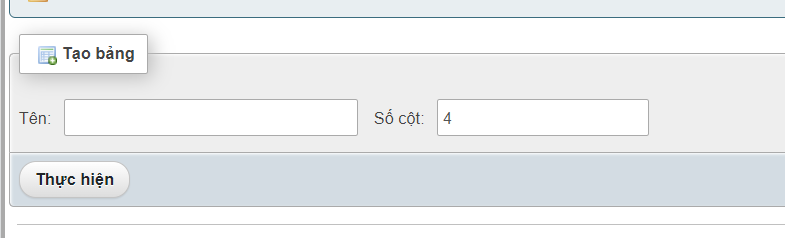
Bấm vào “Mới” để tạo ra một database chứa dữ liệu.

Ảnh có chứa văn bản

Mô tả được tạo tự động

Hình 5:

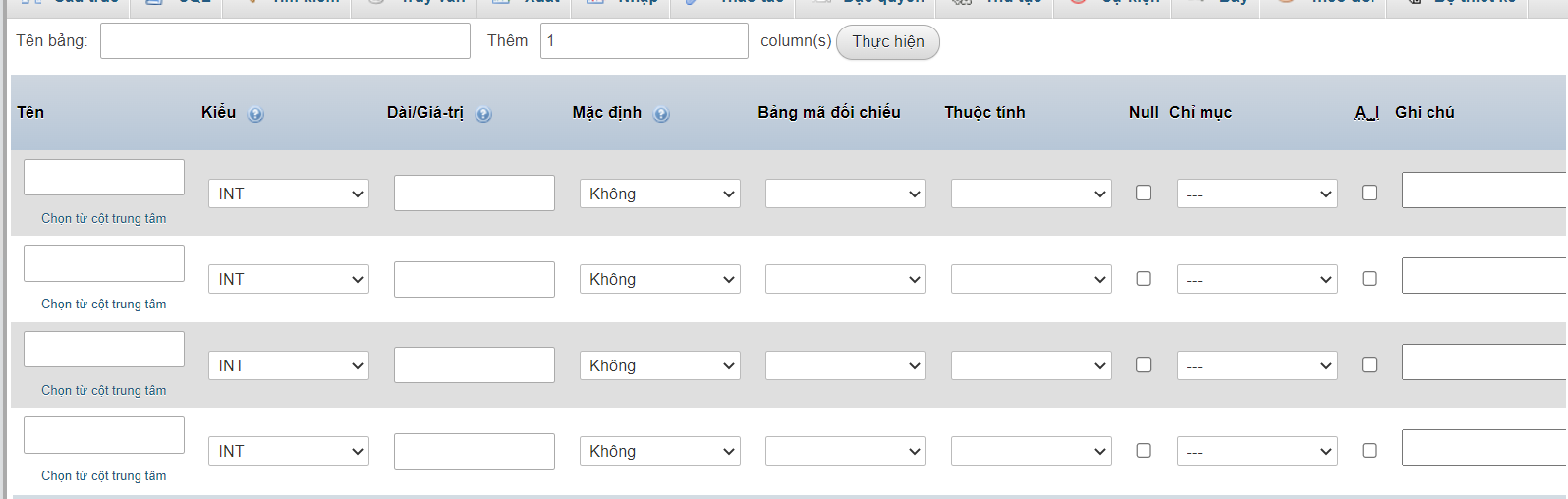
Đặt tên cho database và bấm nút tạo.



Hình 6:

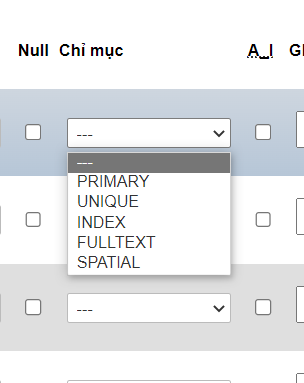
Sau khi đã tạo database, tiếp theo sẽ tạo table.

Gồm có đặt tên và số cột chứa các khóa.



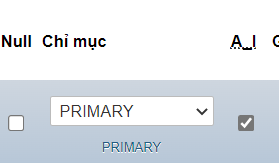
Hình 7:

Khi đã đặt tên và chọn số cột ( Ví dụ : 4), tiếp theo sẽ đặt tên, gán khóa chính, chọn kiểu dữ liệu cần thiết cho khóa, số giá trị (được nhập bao nhiêu kí tự).



Hình 8:

Ở phần chỉ mục, là phần để gán khóa chính (PRIMARY).



Hình 9:

Khi chọn PRIMARY, bấm vào checkbox để lưu khóa chính.

Ảnh có chứa văn bản

Mô tả được tạo tự động

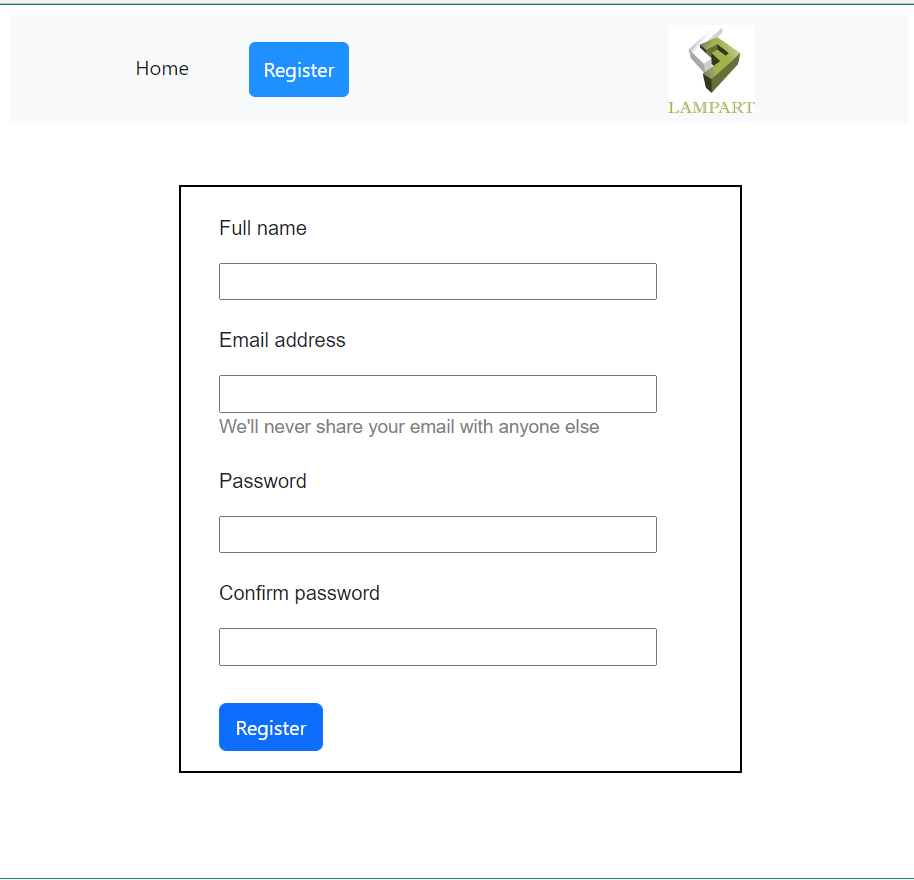
Bấm vào “Ghi lại” để lưu các khóa đã được tạo ra.

Ảnh có chứa văn bản

Mô tả được tạo tự động

Hình 11: Ví dụ khi hoàn thành

1. **Register**



Hình 13: Giao diện đăng nhập

Ảnh có chứa văn bản

Mô tả được tạo tự động

Hình 13 : File Register.php

* Sử dụng <form> để lấy dữ liệu của file này đến file được liên kết nhờ vào action.
* Action là đường nối liên kết.
* Method sẽ gồm 2 loại là POST ( phù hợp khi cho đăng nhập, đăng kí,…) và GET ( phù hợp khi tìm kiếm vì nếu đăng nhập hoặc đăng kí, dữ liệu sẽ bị đẩy lên URL như vậy sẽ dễ dàng bị mất tài khoản).
* Ở các thẻ input, ta có *name = ”… ”* mục đích để chỉ rõ dữ liệu đã nhập vào thuộc phần nào.
* Ở phần thẻ button, cũng có *name = “submit”, khi bấm vào button sẽ nhảy đến link trong action.*



Hình 14:

*Isset($\_POST[‘submit’]* là xem phần name= “submit” thuộc button trong file Register.php có tồn tại (nghĩa là có được bấm vào không) và gán vào *$check.*

Để kiểm tra, ta có thể sử dụng câu lệnh *var\_dump($check)*.

* Nếu không tồn tại bool(false)
* Nếu tồn tại bool(true)

*Ảnh có chứa văn bản

Mô tả được tạo tự động*

Hình 15:

Sử dụng *if* với điều kiện trên để kiểm tra xem *$check* đã được submit hay không, *$\_POST[”name”], $\_POST["email"], $\_POST["password"], $\_POST["repassword"]* có bị để trống hay không.

Nếu *$check* có tồn tại và *$\_POST[”name”], $\_POST["email"], $\_POST["password"], $\_POST["repassword"]* không bị để trống thìgán từng loại theo từng tên khác nhau, tùy theo bản thân muốn đặt gì.

Ảnh có chứa văn bản

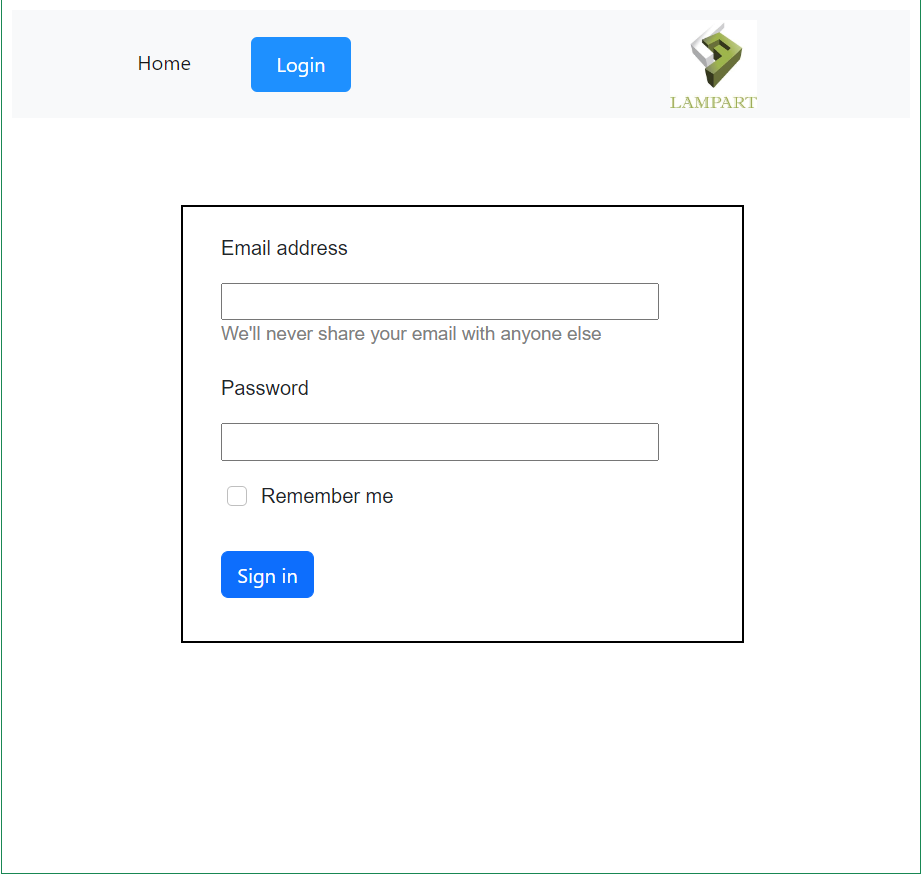
Mô tả được tạo tự động

Hình 16:

*If* trên dùng để xem *$password* có khác với *$repassword* không.

* Nếu khác sẽ quay trở lại file Register.php để người dùng nhập lại password và repassword.
* Nếu giống nhau thì gọi email trong table user với điều kiện là *email* trong table *user* giống với *$email* đã được gán ở trên và sau đó gán vào *$sq.*
  + *Mysqli\_query($config, $sq*) là dùng để truy vấn và database với *$config* thuộc file *config.php* dùng để kết nói với datable và *$sq* đã được gán ở trên.
  + *Mysqli\_num\_rows($check)* dùng để đếm dữ liệu trong database, với $check truy vấn vào database với kiểm tra cột email trong table user với điều kiện *email* trong table *user* giống với *$email.*
* Nếu email giống $email thì *mysqli\_num\_rows($check)* sẽ bằng 1.
* Nếu không giống thì *mysqli\_num\_rows($check)* sẽ bằng không.
* Và sử dụng if để kiểm tra đã có email trong database chưa.
  + Nếu có sẽ chuyển lại về Register.php.
  + Nếu không có thì sẽ thêm (Insert into ) vào table *user* các cột *username, email, password, level* theo từng giá trị *$username, $email, $password, $level* đã được gán ở trên.
  + Và sử dụng mysqli\_query để truy vấn vào database với hoạt động trên.
  + Sau đó, sẽ chuyển đến file login.php.

1. **Login**

****

Hình 17: Giao diện đăng nhập

Ảnh có chứa văn bản

Mô tả được tạo tự động

Hình 18:

Tương tự như Register.php, có <form> để lấy dữ liệu của file này đến file được liên kết nhờ vào action.

Ảnh có chứa văn bản

Mô tả được tạo tự động

Hình 19:

Tương tự như ở *dki.php*, sẽ có điều kiệnđể kiểm tra xem đã được submit hay không, *$\_POST["email"], $\_POST["password"]* có bị để trống hay không.

Nếu có submit và không để trống thì gán dữ liệu.

Chọn tất cả các cột từ table user với điều kiện trên gán và *$sql* và dùng *mysqli\_query($config, $sql)* để truy vấn.

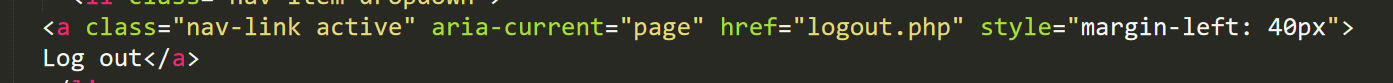
*Mysqli\_fetch\_assoc($query*) là dùng để lấy dữ liệu từ database ra ngoài.

*Session\_start()* là dùng để lưu trữ dữ liệu, thông tin và được sử dụng trên nhiều trang.

Gán *$\_SESSION[“user”] = $data* được gán với *mysqli\_fetch\_assoc($query)* nghĩa là *$\_SESSION[“user”]* sẽ có dữ liệu trong database với điều kiện email = $email, password = $password. Có nghĩa rằng khi người dùng đăng nhập đúng với dữ liệu trong database thì email và password sẽ được lưu vào *$\_SESSION[“user”].*

Tương tự như *$\_SESSION[“email”],* lưu được email mà người dùng đăng nhập.

1. **Logout**

****

Hình 20 : Đường linh dẫn đến logout.php

Khi bấm vào đường link trên, nó sẽ nhảy sang logout.php và *session\_start()* đã lưu dữ liệu từ login.php sẽ bị hủy bỏ và quay trở về home.php

Ảnh có chứa văn bản

Mô tả được tạo tự động

Hình 21:

1. **Home**
2. **Kiểm tra đăng nhập hoặc chưa đăng nhập**

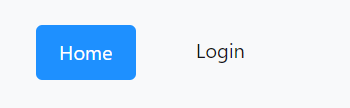
**Ảnh có chứa văn bản

Mô tả được tạo tự động**

Hình 22:

If này là để kiểm tra xem đã đăng nhập hay chưa dựa vào *session* từ *dnhap.php.*

Để xem nếu đã đăng nhập thì email trong session có tồn tại và ngược lại.



Hình 23:

Ảnh có chứa văn bản

Mô tả được tạo tự động

Hình 24:

Nếu chưa đăng nhập, thì nút kế bên home sẽ là Login và Register.

Ảnh có chứa văn bản

Mô tả được tạo tự động

Hình 25:

Ảnh có chứa văn bản

Mô tả được tạo tự động

Hình 26:

Nếu đã đăng nhập thì nút kế bên Home sẽ là Log out.

1. **Phân trang**

****

Hình 27:trong file admin.php

Gán *$page\_s1* là gọi tất cả cột trong table user gộp nhóm qua id với giới hạn tăng dần là từ *$begin* đến 3



Hình 28:

Khi bấm vào trang 1, Phần URL sẽ hiển thị số trang nhờ vào câu lệnh *href="home.php? trang= <?php echo $i; ?> .*

Ảnh có chứa văn bản

Mô tả được tạo tự động

Hình 29:

Sử dụng câu lệnh trên để xem trang có tồn tại không

* Nếu không sẽ gán là rỗng.
* Nếu có sẽ gán *$page = $\_GET[‘trang’]* (nghĩa là lấy số $i)



Hình 31 : Trang số 2

Hình 30: trang số 1

Ảnh có chứa văn bản

Mô tả được tạo tự động

Hình 32:

*Ceil($pages/3)* là khi chia *$pages* được gán với *mysqli\_num\_rows($sql)* để xem có bao nhiêu sản phẩm và lấy chia 3 ra kết quả, rồi dùng *ceil* làm tròn kết quả để ra số trang cần thiết.

Ảnh có chứa văn bản

Mô tả được tạo tự động

Hình 33:

If trên dùng để xem $page có bằng rỗng hoặc bằng 1 hay không.

* Nếu bằng thì gán $begin = 0
* Nếu không bằng gán $begin = ($page \* 3)-3
* Ví dụ: $page đang bằng 2 thì lấy 2 nhân 3 và trừ 3 ( do lấy 3 là giới hạn dữ liệu ở hình 27) thì sẽ bằng 3, nghĩa là bắt đầu từ dữ liệu có thứ tự là 3 và tăng dần thêm 2 thứ tự.

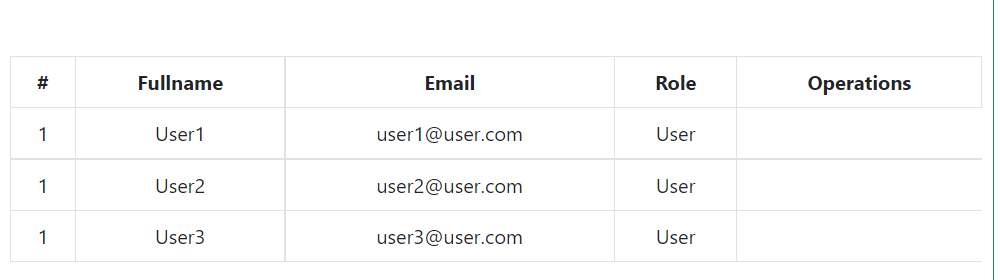
1. **Nonuser**

**Ảnh có chứa văn bản

Mô tả được tạo tự động**

Hình 34: trong file home.php

Tiếp tục dùng if để xem đã đăng nhập hay chưa.



Hình 35: Khi chưa đăng nhập

**Ảnh có chứa văn bản

Mô tả được tạo tự động**

Hình 35 :

Trong nonuser.php, sẽ có phần *SELECT* như **VI.Home (2)** để giới hạn dữ liệu người dùng khi xuất ra ngoài.

Sử dụng vòng lặp *while( $row = mysqli\_fetch\_assoc($sql))* nghĩa là xuất dữ liệusử dụng *$row* vì đã được gán với *mysqli\_fetch\_assoc($sql),* giúp lấy dữ liệu từ database ra ngoài trong khi *$sql* thì truy vấn vào database.

Sử dụng câu lệnh *If($row[‘level’] !=1)* nghĩa là nếu trong cột level có dữ liệu nào có level khác 1 thì sẽ xuất ra ngoài còn nếu bằng 1 thì sẽ không xuất.

Vì đã mặc định gán dữ liệu admin có level = 1, nhằm để tránh nhầm lẫn.

Khi chưa đăng nhập sẽ chỉ có hiển thị thông tin người dùng và không có chức năng sửa, xóa, copy, xem chi tiết.

Ảnh có chứa văn bản

Mô tả được tạo tự động

Hình 36: trong home.php

1. **Admin**

**Ảnh có chứa văn bản

Mô tả được tạo tự động**

Hình 37:

**Ảnh có chứa văn bản

Mô tả được tạo tự động**

Hình 38: Khi đăng nhập vào tài khoản Admin

**Ảnh có chứa văn bản

Mô tả được tạo tự động**

Hình 39:

Tương tự như nonuser với phần *SELECT* như **VI.Home (2)** để giới hạn dữ liệu người dùng khi xuất ra ngoài.

Sử dụng câu lệnh *If($row[‘level’] ==1)* để xem nếu là Admin sẽ có Role là Admin, nếu là user sẽ có role là user.

Và Admin có chức năng sửa, xem , xóa danh sách người dùng User.

1. **User**

**Ảnh có chứa văn bản

Mô tả được tạo tự động**

Hình 40:

**Ảnh có chứa văn bản

Mô tả được tạo tự động**

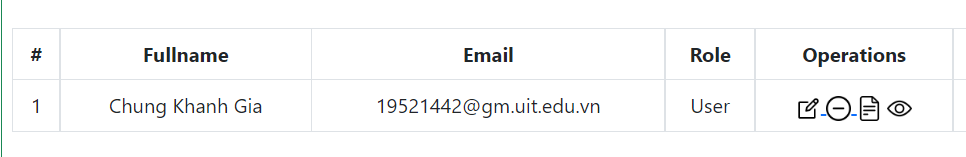
Hình 41:

Khác với nonuser và admin, user sẽ sử dụng session để chỉ xuất ra mỗi dữ liệu đăng nhập của người dùng và kèm theo chức năng sửa, xóa, copy, xem chi tiết.

*(isset($\_SESSION['user'])) ? $\_SESSION['user'] : []* câu lệnh này dùng để xem *session[‘user’]* có tồn tại hay không và gán vào $user.

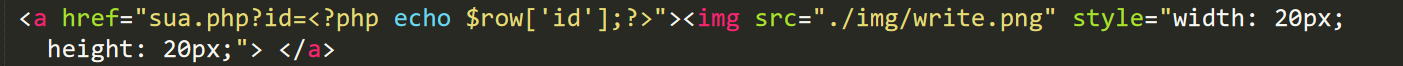
Qua đó, dữ liệu lưu trong session[‘user’] được gán vào $user.

Ví dụ nếu gọi *$user[‘username’]* thì sẽ gọi đến tên username lấy từ database ra khi đăng nhập.

****

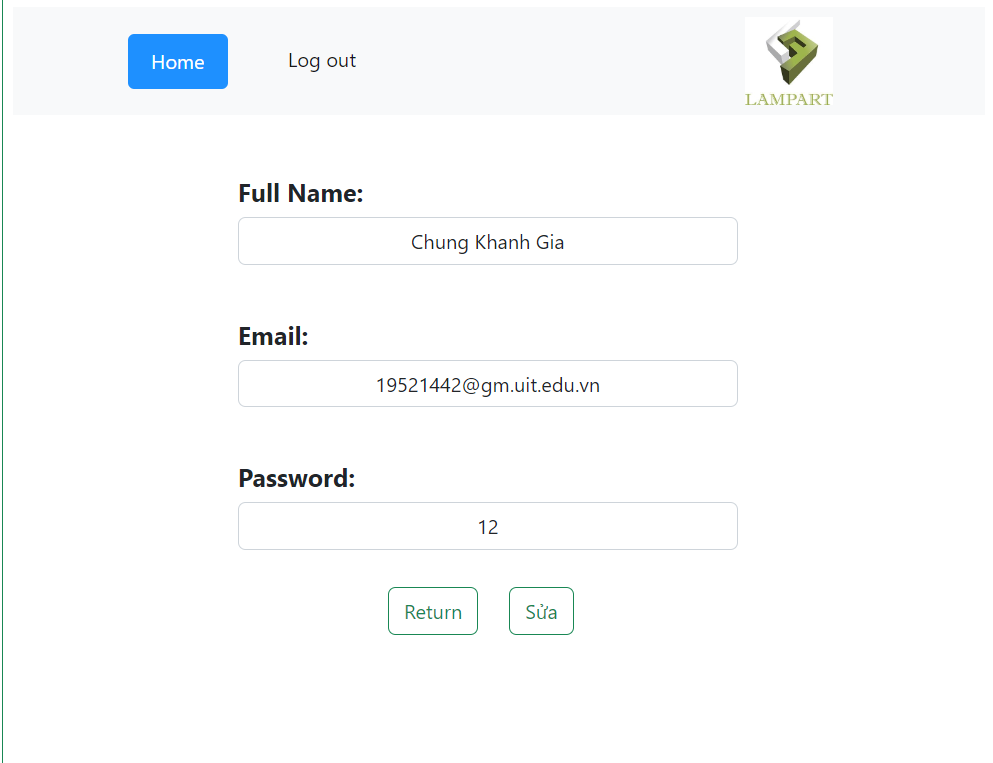
Hình 42: Khi đăng nhập vào tài khoản User

1. **Sửa**

****

Hình 43: Link dẫn đến sua.php

Trong thẻ <a> có dòng *sua.php?id=<?php echo $user['id'];?>* nghĩa là idsẽ được xuất ra trên URL.



Hình 44: Giao diện sửa

Giao diện sửa hiển thị tất cả thông tin của người dùng.

**Ảnh có chứa văn bản

Mô tả được tạo tự động**

Hình 45 :

Ảnh có chứa văn bản

Mô tả được tạo tự động

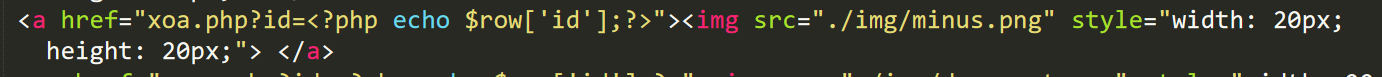
Hình 46:

Sau đó, dùng if để xem id được lấy từ URL nhờ vào $\_GET[“id”] có tồn tại hay không.

Khi bấm vào đường link ở hình 43 sẽ chuyển đến trang sua.php, nơi có *<form method = “POST”>* và trong các thẻ *input* sẽ xuất ra dữ liệu có điều kiện *id = $id.*

Và khi sửa lại dữ liệu trong input, bấm vào nút sửa thì sẽ cập nhật (update) lại user với *username = '$username', email = '$email', password = '$password'* và điều kiện *id = $id.*

1. **Xóa**

****

Hình 47: Link dẫn đến xoa.php

Tương tự như sua.php, sử dụng if để xem id được lấy từ URL nhờ vào *$\_GET[“id”]* có tồn tại hay không.

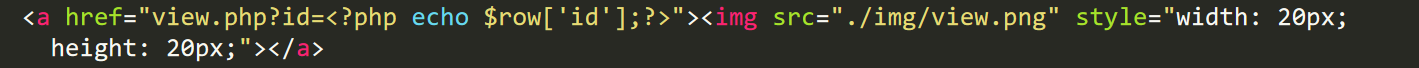
Nếu có thì sẽ xóa tất cả từ table user với điều kiện *id = $id* và dùng mysqli\_query để truy vấn vào database.

**Ảnh có chứa văn bản

Mô tả được tạo tự động**

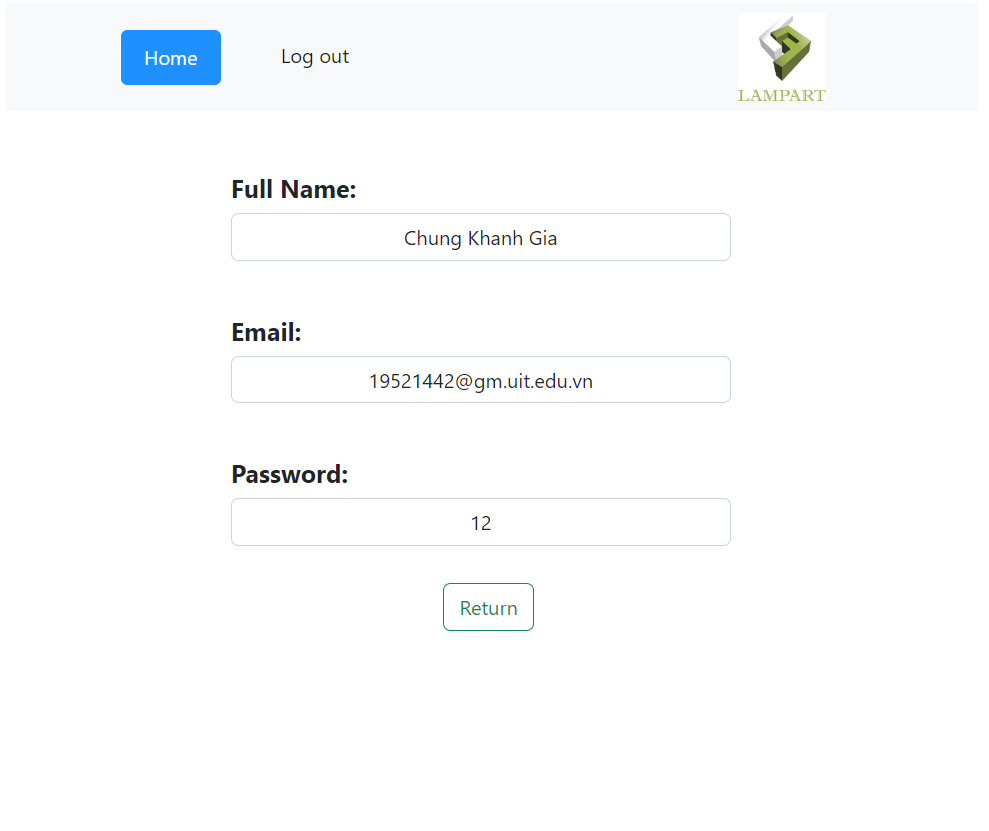
Hình 48:

1. **Xem chi tiết**

****

*Hình 49: Link dẫn đến view.php*

Trang view.php giống với sua.php về phần giao diện nhưng sẽ không cập nhật được dữ liệu trong đó.

**

Hình 50: Giao diện xem chi tiết