**Wirte up PicoCTF**

# *WEBEXPLOITATION*

# WebDecode

A screenshot of a computer

Description automatically generated

Nhìn tên chall có liên quan đến Decode, thử f12 thì thấy 1 chuỗi base64 encoded cGljb0NURnt3ZWJfc3VjYzNzc2Z1bGx5X2QzYzBkZWRfMTBmOTM3NmZ9, khi decode thì ta có flag:

Flag: picoCTF{web\_succ3ssfully\_d3c0ded\_10f9376f}

# Unminify

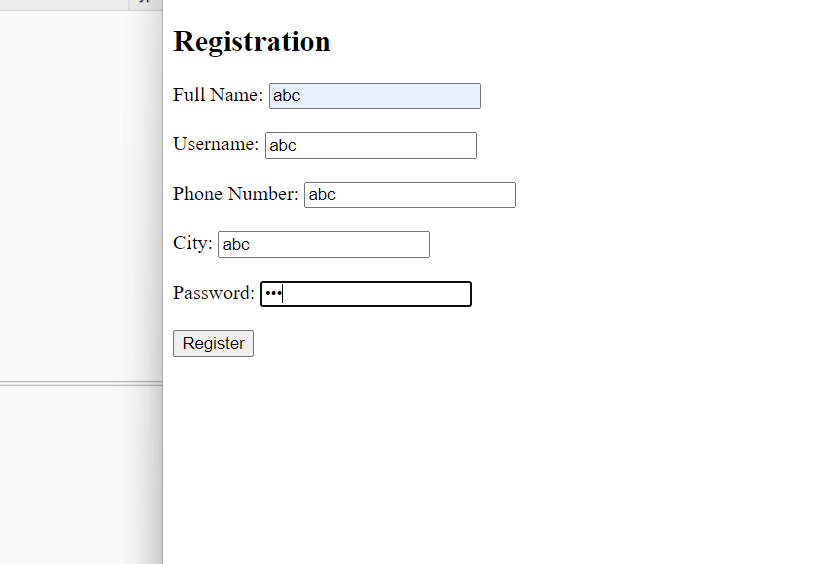
A screen shot of a computer

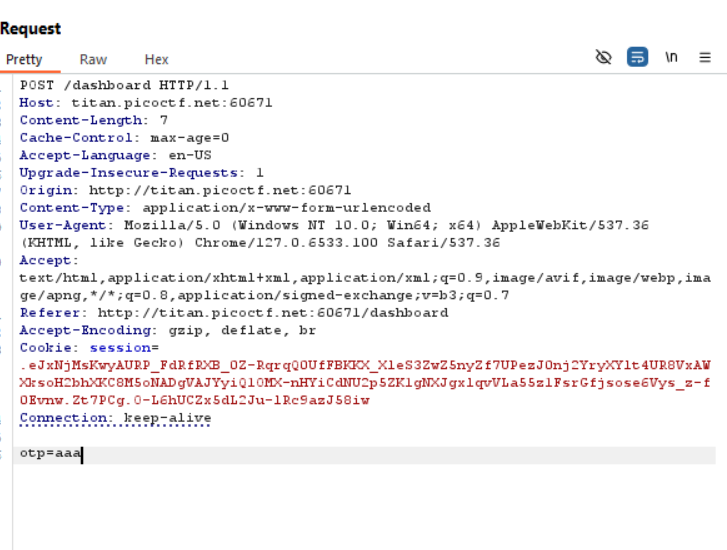
Description automatically generated

F12 ra thấy luôn flag là class của thẻ p luôn 😉

Flag: picoCTF{pr3tty\_c0d3\_d9c45a0b}

# IntroToBurp





Thử điền bất kỳ dữ liệu gì vào các ô input và bấm Register thì đều gửi mã OTP, Thử nhập chuỗi số ngẫu nhiên vào ô input, bấm Submit và kiểm tra bằng Burpsuite thì thấy Header của request có chứa trường otp= “chuỗi số vừa nhập”. Xóa dòng này đi và send lại request ta nhận được flag.

A screenshot of a computer

Description automatically generated

Flag: picoCTF{#0TP\_Bypvss\_SuCc3$S\_b3fa4f1a}

# Bookmarklet

A screenshot of a computer

Description automatically generated

Mở chall ta thấy đoạn code JS

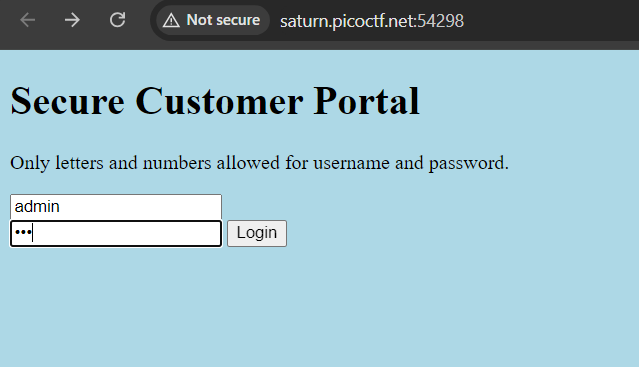
A screen shot of a computer screen

Description automatically generated

Chạy đoạn code Js hiện ở Bookmarklet , Chương trình sẽ alert ra nội dung flag:

Flag: picoCTF{p@g3\_turn3r\_1d1ba7e0}

# Local Authority



A computer screen shot of a program code

Description automatically generated

F12 mở soure code có file secure.js, đọc code có thể thấy nếu uername bằng ‘admin’ và password bằng ‘strongPassword098765’ thì sẽ trả về true.

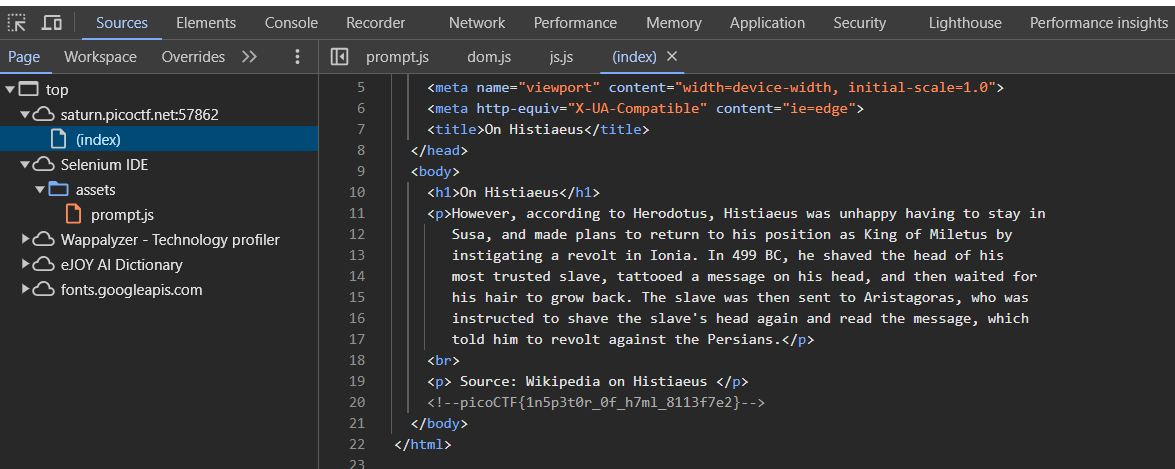
A blue screen with black text

Description automatically generated

Copy tài khoản đó rồi đăng nhập ta sẽ nhận flag:

Flag: picoCTF{j5\_15\_7r4n5p4r3n7\_a8788e61}

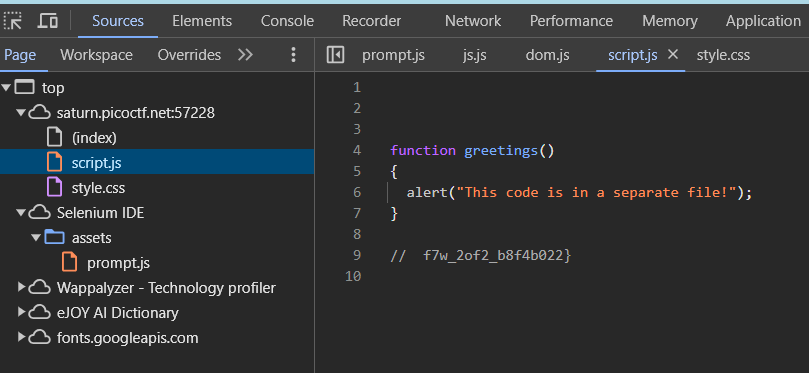
# Inspect HTML



Như tên chall, Inspect trang web ra ,chọn phần Sources trong file index sẽ có flag nằm trong phần comment

Flag: picoCTF{1n5p3t0r\_0f\_h7ml\_8113f7e2}

# Includes



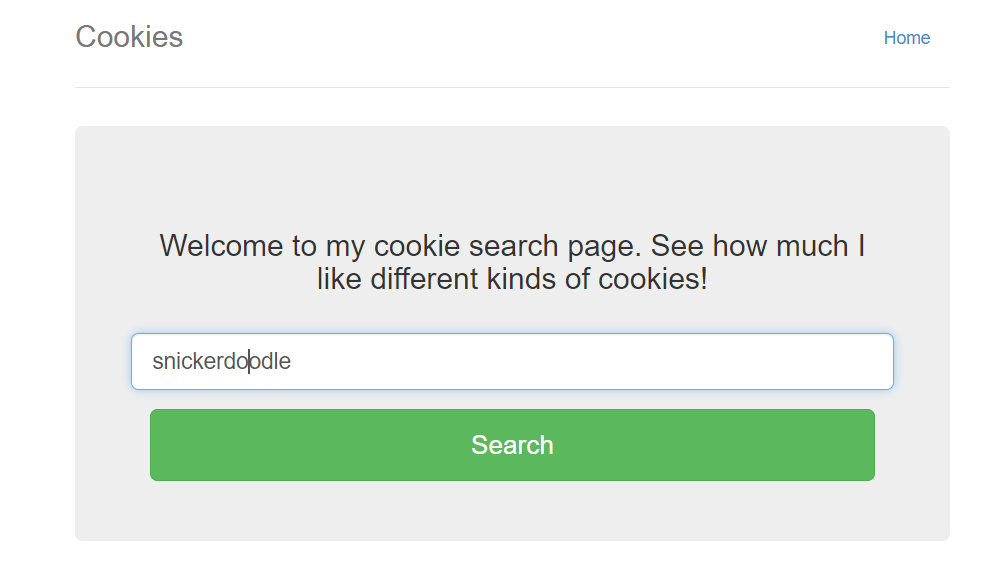
A screenshot of a computer program

Description automatically generated

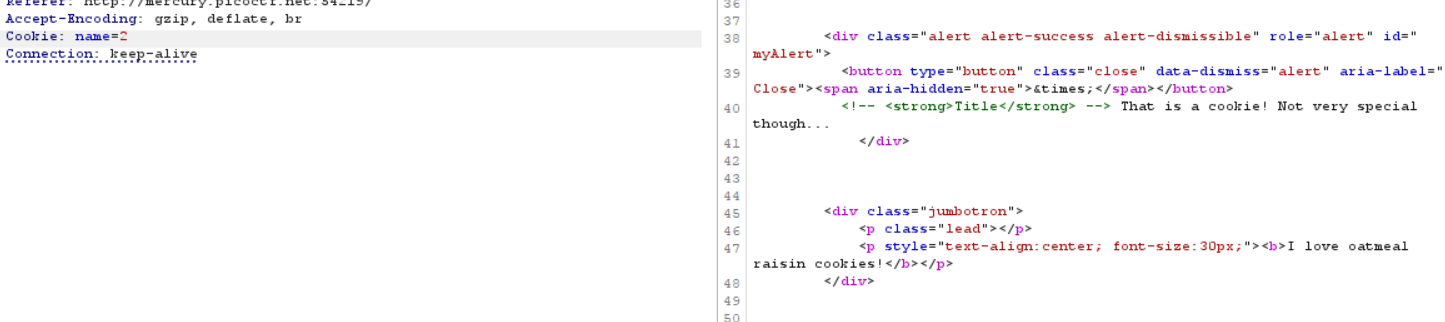
Inspect trang web ta sẽ thấy trong phần Sources sẽ có 2 file là Script.js và style.css. Trong file script.js sẽ có phần đuôi của flag và file Style là phần đầu, ghép lại ta sẽ có flag full.

Flag: picoCTF{1nclu51v17y\_1of2\_ f7w\_2of2\_b8f4b022}

# Cookies



Trong thanh input có phần placehoder là “snickerdoodle”, điền chuỗi này vào ô input và Search thử thì thấy submit thành công nhưng chưa nhận được flag. Tên chall cho thấy bài này liên quan đến Cookie, Mở tab Application trong Inspect hoặc dùng Burpsuite để check.



Ở đây ta dùng Burpsuite, khi submit thành công ta thấy cookie có value là 0, thử đổi thành số khác thì thấy web trả về tên 1 loại bánh khác , thử lần lượt các số và xem sự thay đổi, mình thử đến số 18 là thấy flag

A screenshot of a computer

Description automatically generated

Flag: picoCTF{3v3ry1\_l0v3s\_c00k135\_96cdadfd}

# Scavenger Hunt

Tìm từng mảnh của flag để hợp thành 1 flag full, mảnh đầu nằm trong file myjs.js, mảnh 2 nằm ở file mycss.css mảnh thứ 3 nằm ở file robots.txt mảnh thứ 4 nằm ở file .htaaccess và mảnh cuối cùng nằm trong file.DS\_Store.

**robots.txt**:

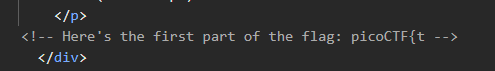
* Đây là một file văn bản nằm trong thư mục gốc của một trang web. Nó được sử dụng để điều khiển cách các công cụ tìm kiếm (như Google, Bing) thu thập dữ liệu trang web. Bằng cách định nghĩa các quy tắc trong file này, quản trị viên trang web có thể cho phép hoặc từ chối các bot truy cập vào một số trang nhất định.

**.htaccess**:

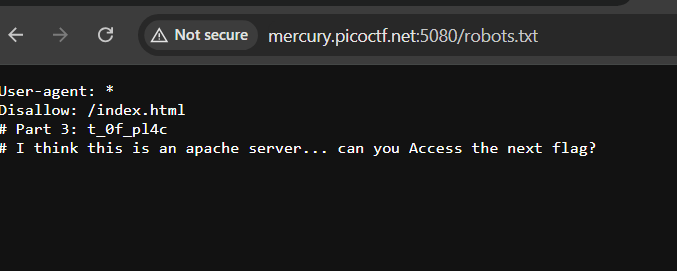
* File .htaccess là một file cấu hình được sử dụng bởi các máy chủ web Apache. Nó cho phép quản trị viên cấu hình nhiều cài đặt như chuyển hướng URL, bảo vệ mật khẩu, điều hướng HTTP sang HTTPS, và ngăn chặn truy cập từ các IP nhất định. File này rất quan trọng cho việc bảo mật và quản lý website.

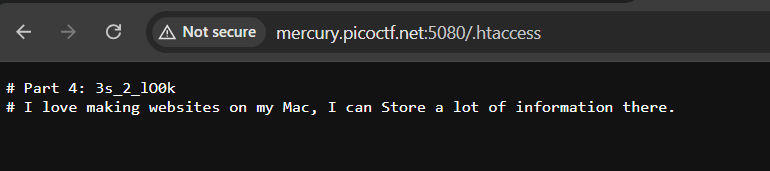
**.DS\_Store**:

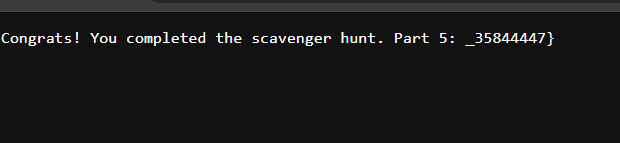
* Đây là một file ẩn được tạo bởi macOS để lưu thông tin về cách hiển thị các thư mục (như vị trí của các biểu tượng, kích thước cửa sổ). Nó không có tác dụng với hệ thống khác nhưng có thể vô tình bị tải lên khi bạn tải thư mục lên máy chủ. File này thường không liên quan đến bảo mật hay quản lý web và có thể bị xóa an toàn khi không cần thiết.



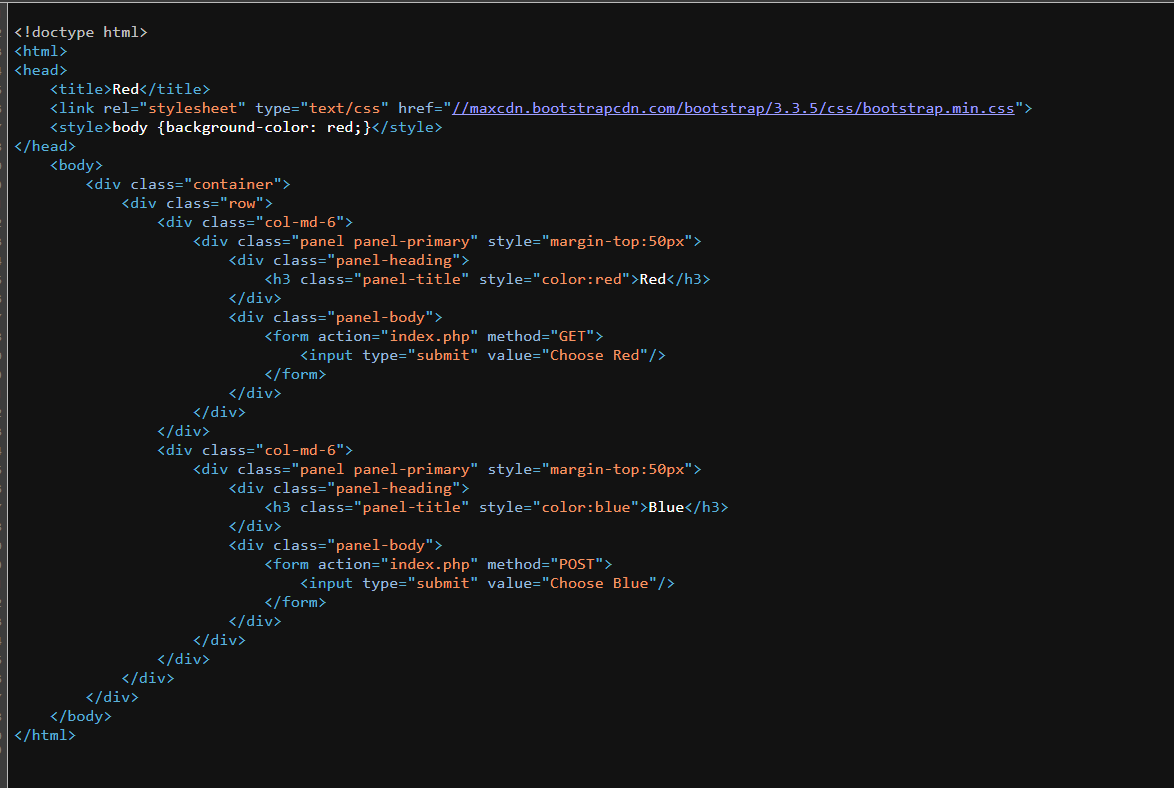








# GET aHEAD



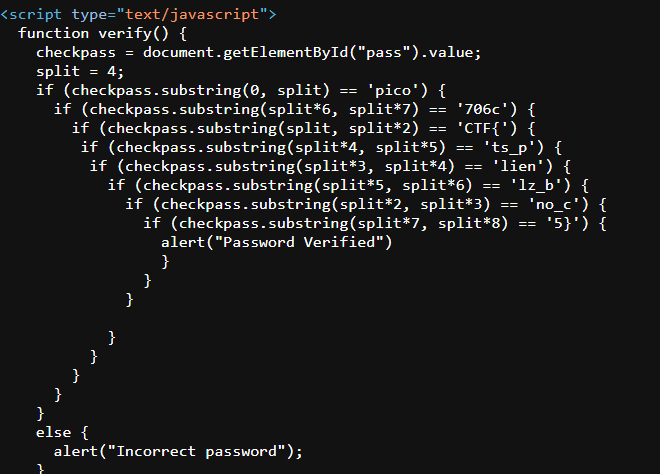
Đọc source ta thấy có 2 giao thức được sử dụng là GET và POST, tên chall viết hoa tất cả chữ HEAD là có ẩn ý, thử đổi GET thành HEAD và send request :

HEAD: Phương thức HEAD trong HTTP cũng được sử dụng để yêu cầu chỉ phần "Header" mà không lấy nội dung thực tế của trang.

curl -I HEAD -i <http://mercury.picoctf.net:53554/index.php>

sau khi send ta sẽ nhận flag

# dont-use-client-side

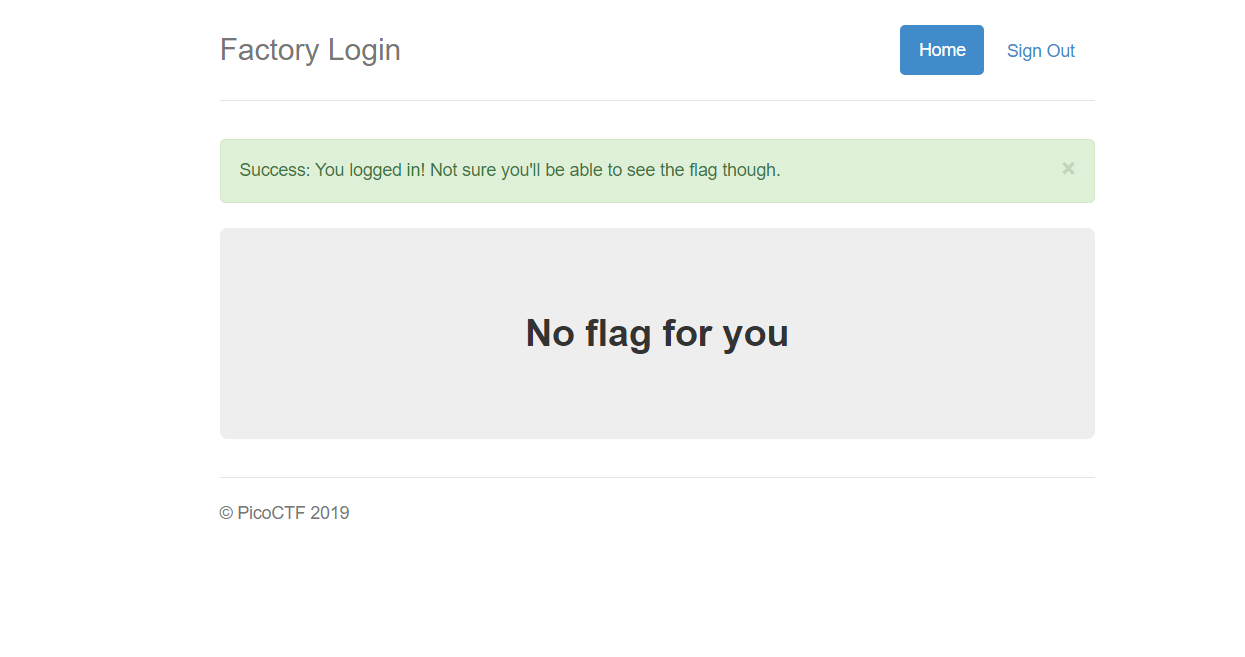


Mở source code và sắp xếp các phần của flag vào thành flag full.

Flag: picoCTF{no\_clients\_plz\_706c5}

# Logon

Đăng nhập thành công bằng acc ‘admin’ , ‘admin’



Inspect xem Cookie ta thấy có 1 cookie admin có value là False, thử đổi thành True rồi load lại trang thì nhận flag

A close-up of a computer code

Description automatically generated

Flag: picoCTF{th3\_c0nsp1r4cy\_l1v3s\_0c98aacc}

# Insp3ct0r

A screenshot of a computer

Description automatically generated

Inspect và trong 3 file này sẽ có 3 phần của flag

# where are the robots

A screenshot of a computer

Description automatically generated

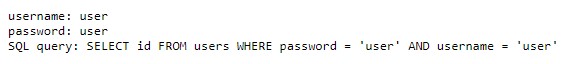
Mở file robots.txt ta thấy có nội dung trên, folow theo path trên ta được flag

A screen shot of a computer

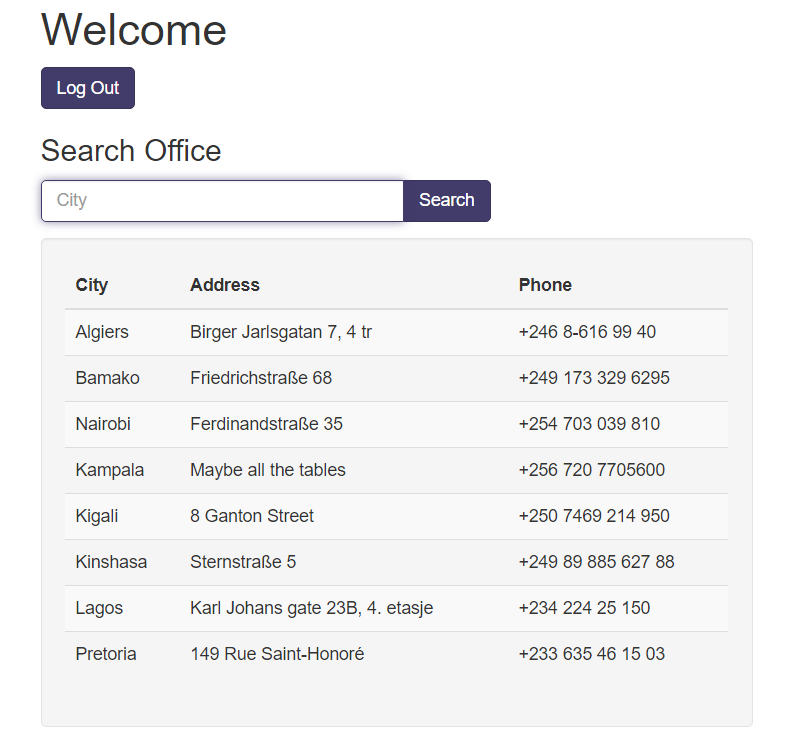
Description automatically generated

# More SQLi

Thử đăng nhập bằng ‘user’ và ‘user thấy có lỗi hiển thị:



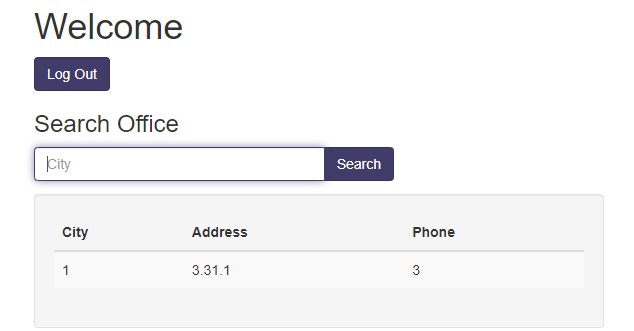
Thử fuzz 'or 1=1;-- vào ô input thì đã bypass thành công



Ta sẽ đi kiểm tra xem thông tin cơ sở dữ liệu , sử dụng loại DB nào

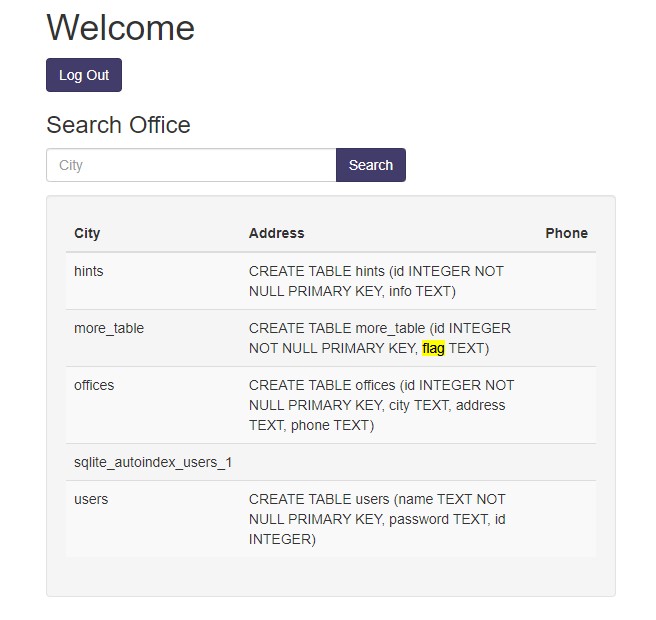
123' UNION SELECT 1, sqlite\_version(), 3;--

Ta có thể thấy web sử dụng SQLite



Tiếp theo là list tất cả các tables ra

123' UNION SELECT name, sql, null from sqlite\_master;--



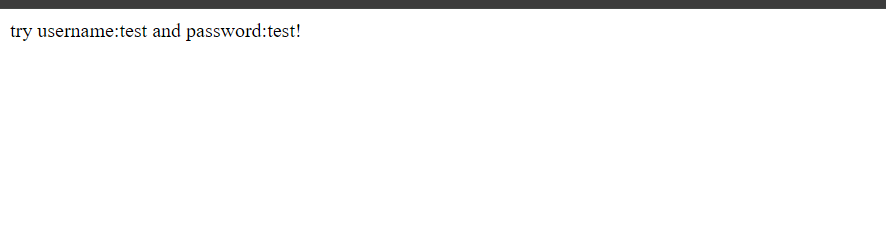
Ta thấy có flag, lấy nó thôi

123' UNION SELECT flag, null, null from more\_table;--

A screenshot of a computer

Description automatically generated

# Findme



Đăng nhập tài khoản là test-test

Khi đăng nhập tài khoản đã có 1 số chuyển hướng rất nhanh, dùng burpsuite để kiểm tra thì thấy 1 respone chuyển tiếp respone login

A screenshot of a computer

Description automatically generated

Login

A screenshot of a computer

Description automatically generated

Login page respone

A screenshot of a computer code

Description automatically generated

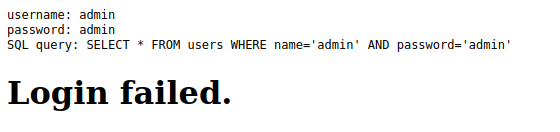
Respone chuyển tiếp sau responr login

ta có thể thấy id= 1 chuỗi base64, encode chuỗi ra ta được flag

flag: picoCTF{proxies\_all\_the\_way\_25bbae9a}

# SQLiLite

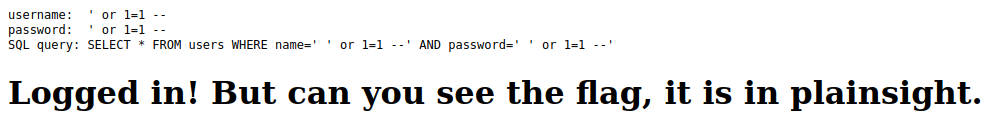
* Chúng ta có thể thấy SQL"i"Lite, trong đó chữ cái i cho chúng ta biết rằng đây có thể là thao tác chèn sql.
* sử dụng tên người dùng=admin và mật khẩu=admin

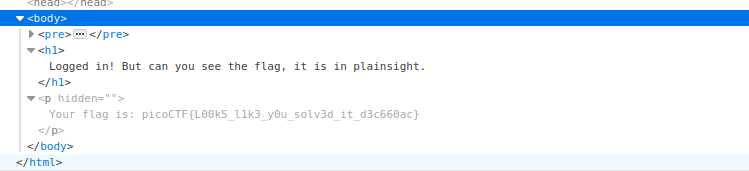
[](https://github.com/fadelmuharam/picoCTF22/blob/master/SQLiLite/images/Screenshot_2022-05-18_23-48-52.png)

• Bây giờ chúng ta đã biết truy vấn được sử dụng để trả về dữ liệu từ bảng.

• Fuzz payload sau

' or 1=1 --

[](https://github.com/fadelmuharam/picoCTF22/blob/master/SQLiLite/images/Screenshot_2022-05-18_23-58-10.png)

* Đã bypass được, giờ hãy Inspect để kiểm tra và lấy flag[](https://github.com/fadelmuharam/picoCTF22/blob/master/SQLiLite/images/Screenshot_2022-05-19_00-01-45.png)