

**TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
KHOA CÔNG NGHỆ THÔNG TIN**



ĐỒ ÁN THỰC HÀNH MẠNG MÁY TÍNH

SOCKET

Thành phố Hồ Chí Minh, tháng 12 năm 2022

**TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
KHOA CÔNG NGHỆ THÔNG TIN**



**ĐỀ 01
WEB SERVER**

**Giáo viên hướng dẫn: Nguyễn Thanh Quân
Lê Hà Minh**

Nhóm: 21120570_21120574_21120580

Thành phố Hồ Chí Minh, tháng 12 năm 2022

MỤC LỤC

I.	LỜI CAM ĐOAN.....	4
II.	LỜI CẢM ƠN.....	5
III.	THÔNG TIN NHÓM.....	6
IV.	ĐÁNH GIÁ MỨC ĐỘ HOÀN THÀNH ĐỒ ÁN.....	7
V.	KỊCH BẢN GIAO TIẾP CỦA CHƯƠNG TRÌNH.....	9
	A. Giao thức trao đổi giữa server và client.....	9
	B. Xử lý thông điệp.....	11
VI.	MÔI TRƯỜNG LẬP TRÌNH VÀ CÁC FRAMEWORK HỖ TRỢ.....	14
VII.	HƯỚNG DẪN SỬ DỤNG CÁC TÍNH NĂNG CỦA CHƯƠNG TRÌNH.....	15
VIII.	BẢNG PHÂN CHIA CÔNG VIỆC.....	23
IX.	TÀI LIỆU THAM KHẢO.....	24

LỜI CAM ĐOAN

Nhóm xin cam đoan kết quả đạt được trong bài báo cáo là sản phẩm riêng của cả nhóm và không có bất kỳ sự sao chép lại của người khác. Trong toàn bộ nội dung của bài báo cáo, những điều được trình bày hoặc là của cá nhân hoặc là được tổng hợp từ nhiều ý kiến của cả nhóm. Tất cả các tài liệu tham khảo đều có xuất xứ rõ ràng và được trích dẫn hợp pháp.

Nhóm xin hoàn toàn chịu trách nhiệm và chịu mọi hình thức kỷ luật theo quy định cho lời cam đoan của nhóm mình.

LỜI CẢM ƠN

Trước tiên với tình cảm sâu sắc và chân thành nhất, cho phép nhóm chúng em được bày tỏ lòng biết ơn đến tất cả các cá nhân và tổ chức đã tạo điều kiện hỗ trợ, giúp đỡ nhóm chúng em trong suốt quá trình học tập và nghiên cứu đề tài báo cáo này. Trong suốt thời gian từ khi bắt đầu học tại trường Đại học Khoa học Tự nhiên, ĐHQG TP.HCM, nhóm đã nhận được rất nhiều sự quan tâm, giúp đỡ của quý Thầy Cô và bạn bè. Với lòng biết ơn sâu sắc nhất, nhóm xin gửi đến quý Thầy Cô đã truyền đạt vốn kiến thức quý báu cho chúng em trong suốt thời gian học tập tại trường. Nhờ có những lời hướng dẫn, dạy bảo của các thầy cô nên bài báo cáo của nhóm mới có thể hoàn thiện tốt đẹp.

Bài báo cáo này được nhóm chúng em thực hiện trong khoảng thời gian gần 2 tuần. Bước đầu đi vào thực tế của nhóm chúng em còn hạn chế và còn nhiều bỡ ngỡ nên không tránh khỏi những thiếu sót, nhóm chúng em rất mong nhận được những ý kiến đóng góp quý báu của quý Thầy Cô để kiến thức của nhóm chúng em trong lĩnh vực này được hoàn thiện hơn đồng thời có điều kiện bổ sung, nâng cao ý thức của mình.

Nhóm chúng em xin chân thành cảm ơn!

THÔNG TIN VỀ NHÓM

STT	Họ và tên	MSSV
1	Đặng Nguyễn Thanh Tín	21120570
2	Nguyễn Minh Trí	21120574
3	Trần Thị Kim Trinh	21120580

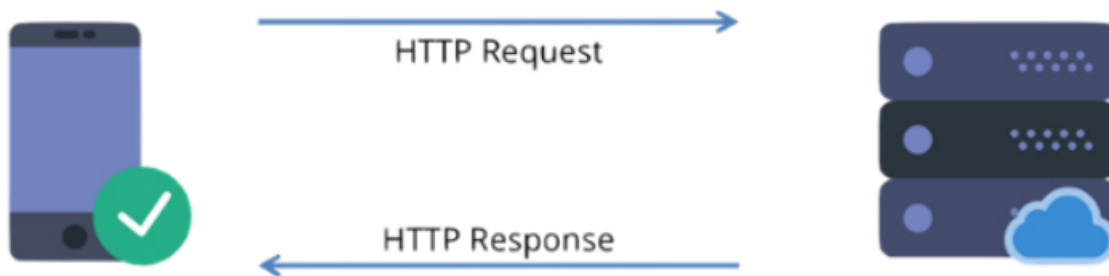
ĐÁNH GIÁ MỨC ĐỘ HOÀN THÀNH ĐỒ ÁN

	Chức năng	Ý nghĩa	Độ hoàn thành	Chú thích
1	Kết nối	Cho phép client kết nối đến server thông qua kết nối TCP	100%	
2	Quản lý kết nối	Khi client hoặc server mất kết nối đột ngột, không làm chương trình treo hay xảy ra lỗi	100%	
3	Tải được page index.html	Browser có thể render lên đầy đủ nội dung của trang index.html. Trang index.html hiển thị 1 HTML form cho phép đăng nhập (yêu cầu 4). Form này có phần action trỏ đến Web server.	100%	
4	Đăng nhập	POST method, gửi " uname " là " admin " và " psw " là " 123456 "	100%	
5	Lỗi page	Trả 404 khi load page không đúng	100%	
6	Multiple requests	gửi nhiều requests trong một connection	100%	
7	Multiple connection	Concurrent, handle nhiều client cùng lúc	100%	

8	Đăng nhập sai tên đăng nhập/mật khẩu	Trả về 401	100%	
---	---	------------	------	--

KỊCH BẢN GIAO TIẾP CỦA CHƯƠNG TRÌNH

A. Giao thức trao đổi giữa Server và Client



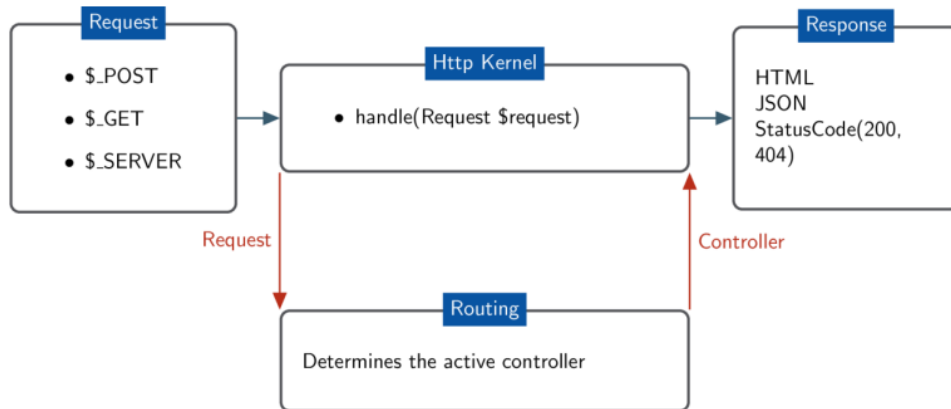
Về phía Client

Máy khách gửi một yêu cầu đến máy chủ một gói dữ liệu gồm phương thức mà máy khách muốn yêu cầu đến máy chủ, **URL** (đường dẫn đến tài nguyên) và phiên bản của giao thức **HTTP**. Tiếp theo đó là các headers (thông điệp MIME) chứa các thông tin về yêu cầu của máy khách, thông tin về máy khách...

Về phía Server

Máy chủ phản hồi với một dòng trạng thái, bao gồm phiên bản giao thức **HTTP**, mã trạng thái (**status code**). Tiếp theo đó là các headers (thông điệp MIME) chứa các thông tin về server và về về nội dung dữ liệu trả về. Cuối cùng là gửi trả lại cho máy khách nội dung mà máy khách yêu cầu (nếu tồn tại).

Định tuyến (Routing)



Khi nhận được một yêu cầu từ máy khách, **server** sẽ xử lý yêu cầu để xác định đường dẫn của tài nguyên mà máy khách muốn truy cập, kiểm tra tài nguyên đó có tồn tại và được quyền truy cập hay không. Nếu tài nguyên tồn tại và được quyền truy cập thì **server** sẽ trả về một gói tin phản hồi bao gồm mã trạng thái kèm với nội dung tài nguyên. Nếu tài nguyên không tồn tại hoặc không được quyền truy cập thì trả về trạng thái **404 (Not Found)**.

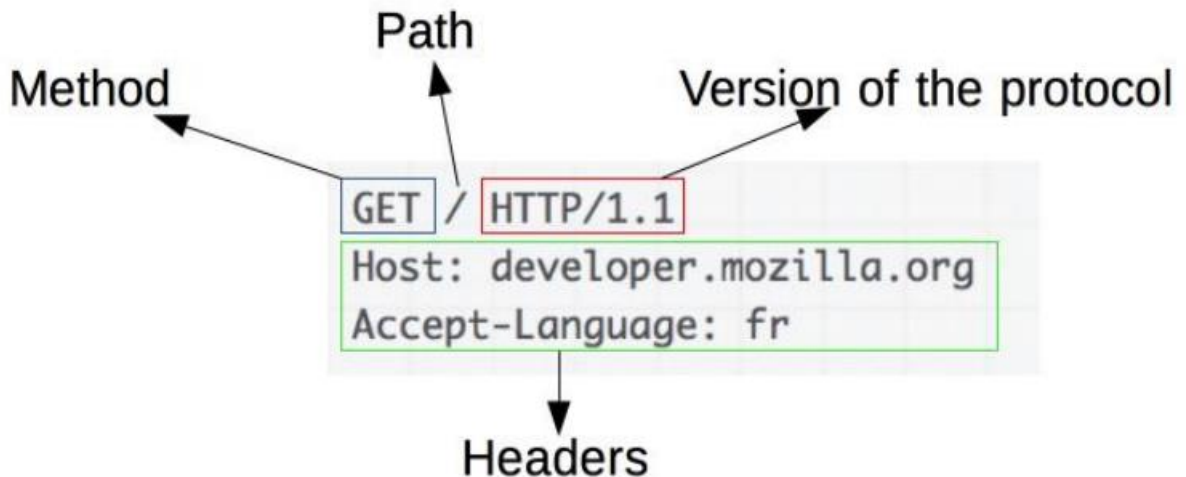
Việc xác định và điều khiển hướng đi cho **Response** là nhiệm vụ của phần định tuyến trong server. Các loại tài nguyên thông dụng được trả về **Content-Type** theo MIME types [1]. Tuy nhiên việc định tuyến được thực hiện tự động dựa trên loại tài nguyên, nhưng cũng có một số ngoại lệ, ví dụ như khi truy cập một đường dẫn, bộ phận định tuyến sẽ kiểm tra xem có tồn tại file nào ở đường dẫn đã yêu cầu và có quyền truy cập hay không, nếu không tìm thấy thì sẽ thử xem đường dẫn đó là một đường dẫn đến thư mục và kiểm tra trong "thư mục" đó có tài nguyên HTML nào hay không, nếu có thì sẽ hiện thị tài nguyên đó. Như vậy bộ phận định tuyến ưu tiên các tài nguyên là file trước, nếu không tìm thấy thì sẽ thử xem đường dẫn đó là một thư mục. Điều này sẽ giúp các truy cập có đường dẫn không rõ ràng như `/` vẫn có thể truy cập bình thường vì lúc này server hiểu đường dẫn chính xác là `/index.html`.

Sau khi đã đăng nhập vào trang website **"/index.html"** thì sẽ được dẫn tới đường dẫn **"/image.html"**

B. Xử lý thông điệp

HTTP Requests

GET Requests



Với các thông điệp yêu cầu, chúng ta cần xác định được phương thức, đường dẫn và phiên bản **HTTP**. **Server** cần phải **parse** thông tin của **request** để biết được cần phải load file nào cho request đó. Nếu là `"/` thì coi như **"index.html"**. Tên filename sẽ liên quan đến thư mục của máy chủ tương ứng (i.e. **css/style.css**, **avatars/1.png**, ...). Sau khi **parse request** để xác định **filename**, **server** sẽ gửi **response** với **HTTP header** tương ứng, kèm theo là **body** (nội dung của file). Kết nối chỉ đóng khi **Client** gửi thông điệp đóng kết nối, do đó máy chủ (**Server**) vẫn phải giữ kết nối (**connection: keep-alive**) cho đến khi **Client** đóng kết nối.

Nếu filename tồn tại, thì trả về thành công "**200 OK**" kèm theo **nội dung của file**:

```
HTTP/1.1 200 OK\r\n
Content-Type: text/html\r\n
Connection: close\r\n
\r\n
<!DOCTYPE html>
<html>
...
```

Nếu file không tồn tại, trả về "**404 File Not Found**"

```
Content-Type: text/html\r\n
Connection: close\r\n
\r\n
<!DOCTYPE html>
<html>
<head>
<title> 404 Not Found </title>
</head>
<body>
<p> The requested file cannot be found. </p>
</body>
</html>
```

POST Requests

Tương tự với **GET Requests**, chúng ta cần đọc các thông tin liên quan. Khác biệt ở chỗ phương thức **POST** sẽ gửi kèm một đoạn dữ liệu để máy chủ xử lý thông tin. Đoạn dữ liệu này được ngăn cách với **header** bằng một chuỗi ngăn cách "**\r\n**".

Server sẽ nhận được thông tin "**uname**" và "**psw**" kèm theo trong body của request. Server cần parse thông tin này ra và kiểm tra nếu "**uname**" là "**admin**" và "**psw**" là "**123456**" thì trả về nội dung của trang "**images.html**". Nếu không thì trả về "**401 Unauthorized**".

```
HTTP/1.1 401 Unauthorized\r\n
Content-Type: text/html\r\n
Connection: close\r\n
\r\n
<!DOCTYPE html>
<h1>401 Unauthorized</h1><p>This is a private area.</p>
```

HTTP Responses

Status Code 200 – OK

Trả về khi yêu cầu của máy khách được máy chủ đáp ứng thành công. Dữ liệu được yêu cầu sẽ trả về kèm theo HTTP Responses.

Status Code 404 – Not Found

Trả về khi máy khách truy cập một tài nguyên không tồn tại, server sẽ trả lời lại bằng một status code 404 để thông báo với client rằng tài nguyên này không thể đáp ứng.

Status Code 401 – Unauthorized

Trả về khi máy khách đăng nhập sai tên tài khoản hoặc là sai mật khẩu (username: "admin", password: "123456"), server sẽ trả lại bằng một status code 401 Unauthorized để thông báo với client rằng mật khẩu đã sai, hãy đăng nhập lại để truy cập.

MÔI TRƯỜNG LẬP TRÌNH VÀ CÁC FRAMEWORK HỖ TRỢ ĐỂ THỰC THI ỨNG DỤNG

A. Môi trường lập trình

- **Môi trường hệ điều hành: Windows**
- **Môi trường phát triển tích hợp(IDE):**
 - **Microsoft Visual Studio Code:** một trình soạn thảo mã nguồn mở gọn nhẹ nhưng có khả năng vận hành mạnh mẽ, là sự kết hợp giữa **IDE** và **CODE Editor**.
- **Ngôn ngữ lập trình được sử dụng để thực hiện đồ án:**
 - **Python:** ngôn ngữ lập trình chính được sử dụng xuyên suốt trong đồ án.

B. Các Framework hỗ trợ

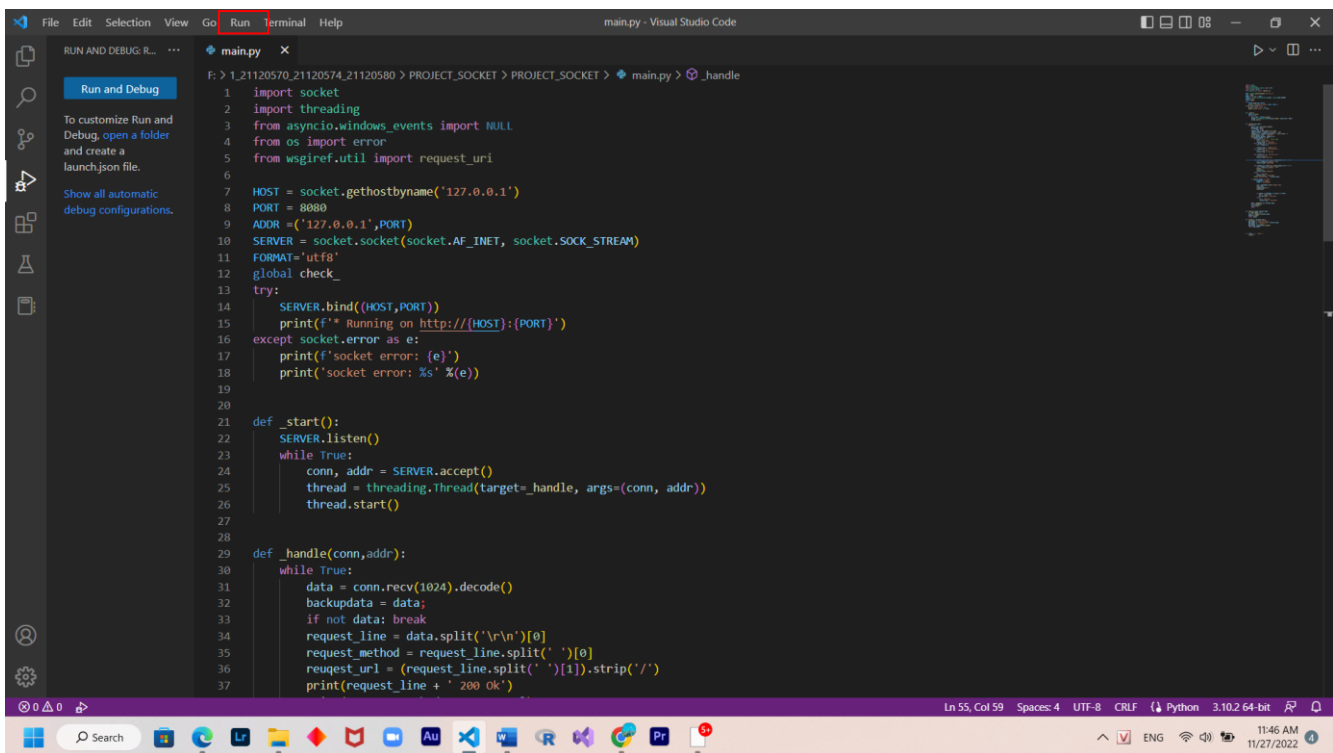
- **Libraries:**
 - **Socket**
 - **Threading**
- **Compilers:**
 - Trình biên dịch ngôn ngữ lập trình **Python** sang ngôn ngữ cho máy trên IDE **Microsoft Visual Studio Code**

HƯỚNG DẪN SỬ DỤNG CÁC TÍNH NĂNG CHƯƠNG TRÌNH

- **Bước 1:** Giải nén file **1_21120570_21120574_21120580**
- **Bước 2:** Vào thư mục **1_21120570_21120574_21120580**, mở tệp tin **main.cpp** bằng IDE **Microsoft Visual Studio Code**

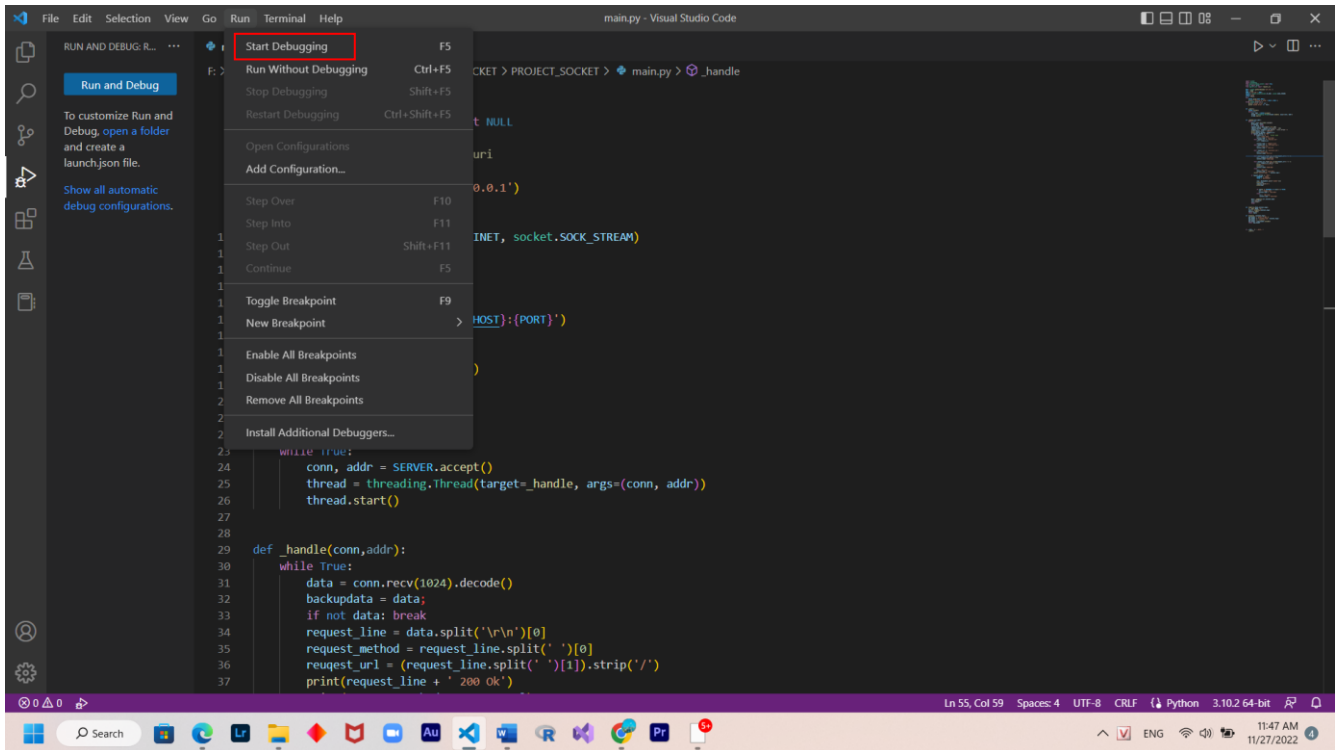
Nếu máy chưa từng cài **Microsoft Visual Studio Code** hay đã có nhưng chưa tích hợp để chạy ngôn ngữ lập trình **Python** thì nhấn vào link để xem hướng dẫn: [NHẤN TẠI ĐÂY](#)

- **Bước 3:** Vào **Run -> Start Debugging** (Hoặc ấn **F5** để run)

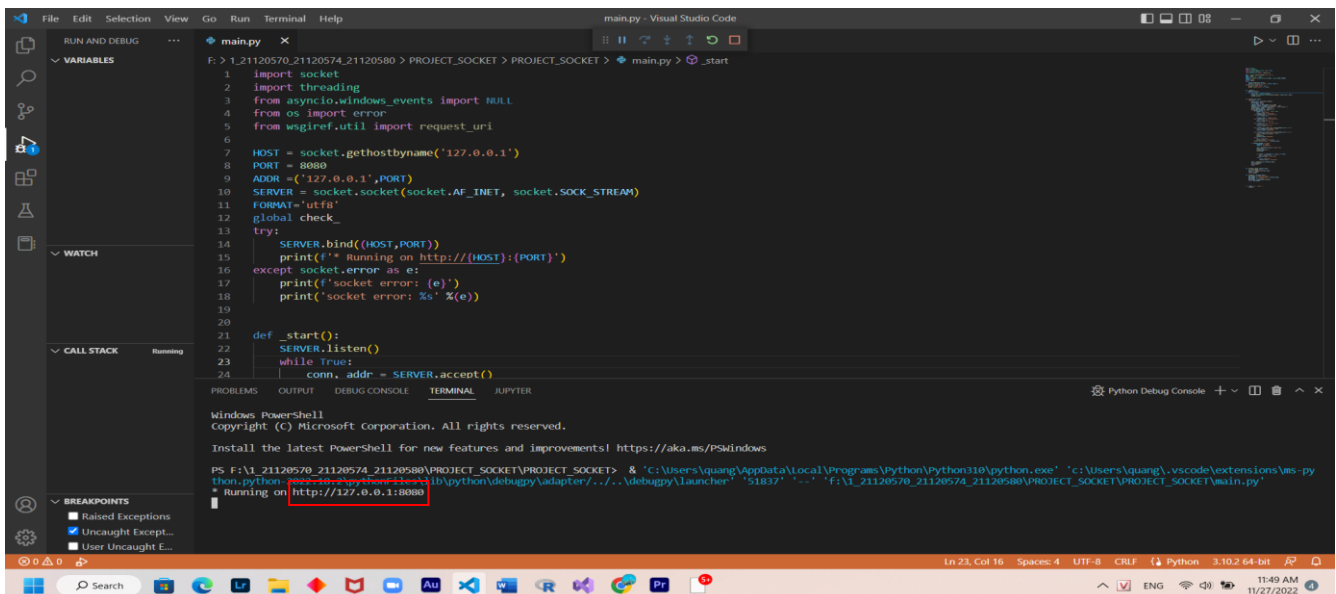


The screenshot shows the Visual Studio Code interface with a Python file named `main.py` open. The code is a simple web server using `socket` and `threading`. The `Run` button in the top menu bar is highlighted with a red box. The left sidebar shows the `Run and Debug` panel. The bottom status bar indicates the file is at line 55, column 59, using UTF-8 encoding and CRLF line endings, with the Python 3.10.2 64-bit interpreter selected.

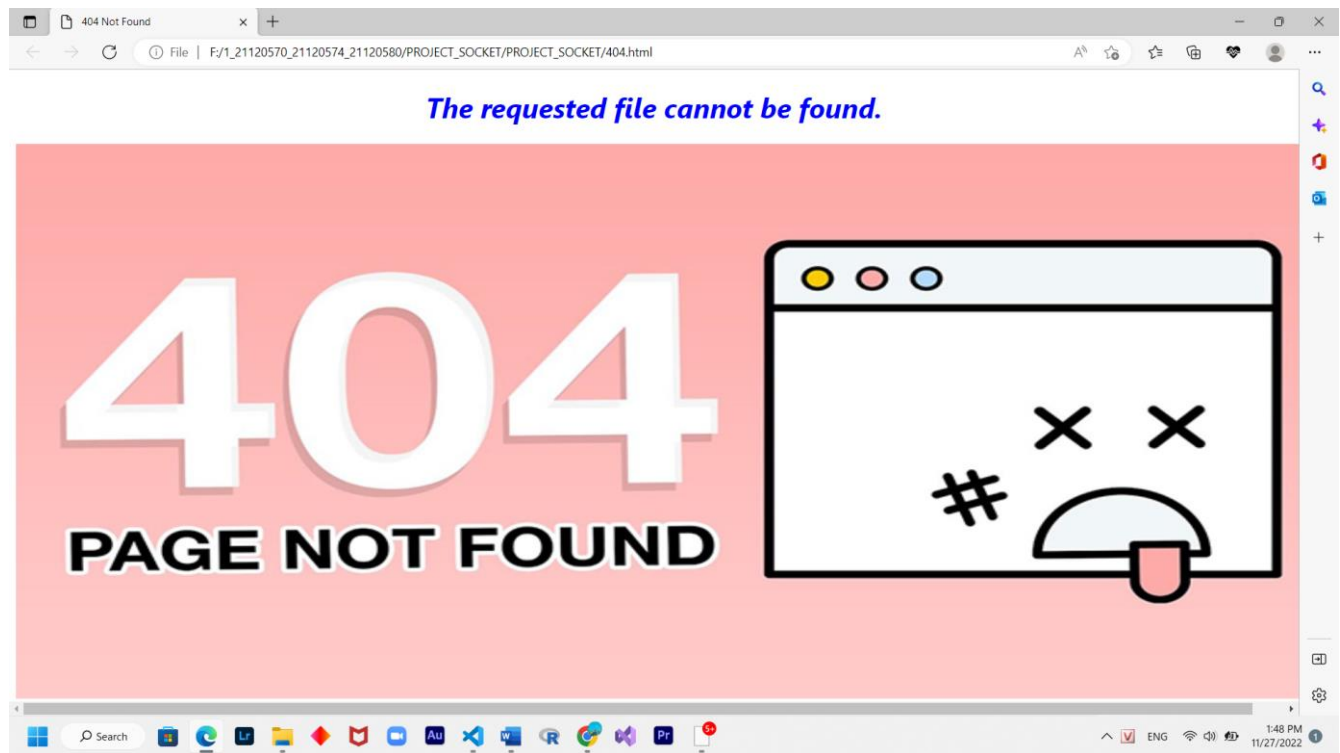
```
1 import socket
2 import threading
3 from asyncio.windows_events import NULL
4 from os import error
5 from wsgiref.util import request_uri
6
7 HOST = socket.gethostbyname('127.0.0.1')
8 PORT = 8080
9 ADDR = ('127.0.0.1', PORT)
10 SERVER = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
11 FORMAT='utf8'
12 global check_
13 try:
14     SERVER.bind((HOST,PORT))
15     print(f'* Running on http://{HOST}:{PORT}')
```



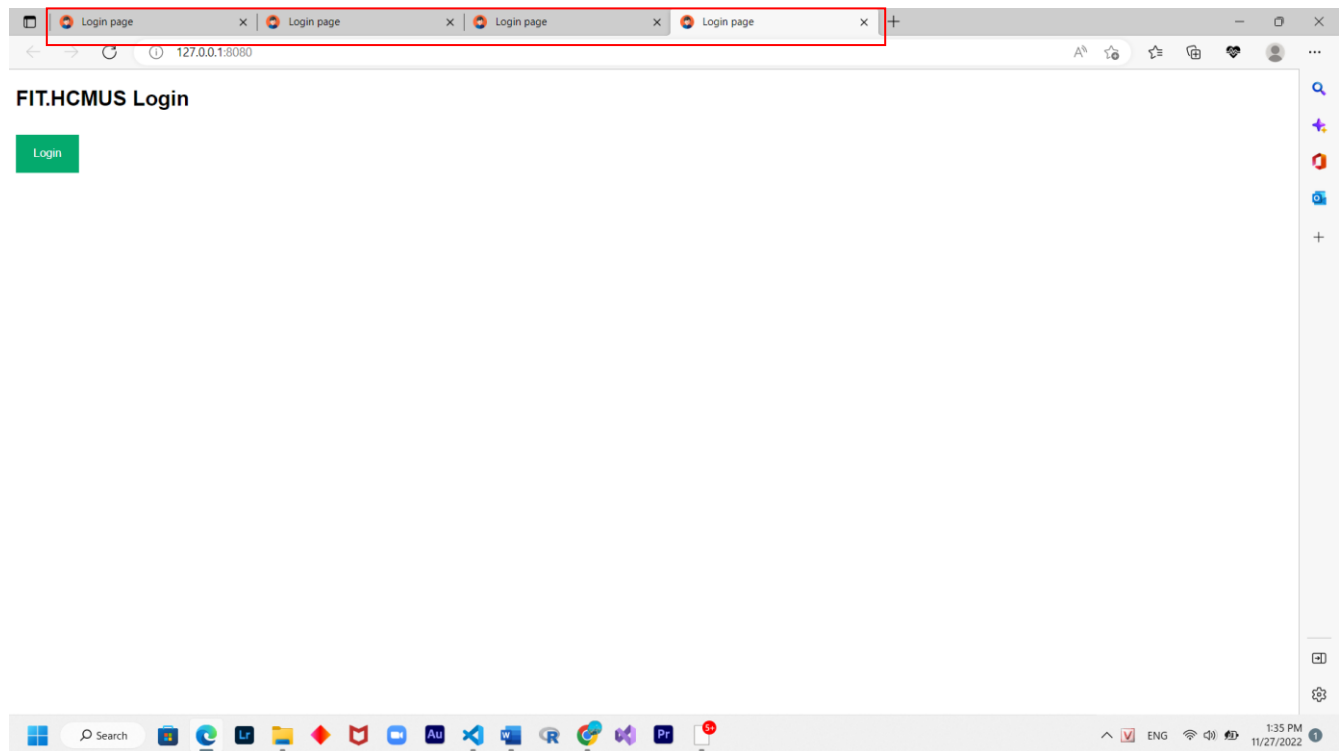
- **Bước 4:** Sau khi chương trình chạy:
Nhấn giữ **phím CTRL** và click chuột vào dòng chữ <http://127.0.0.1:8080> trên dòng *
Running on <http://127.0.0.1:8080>



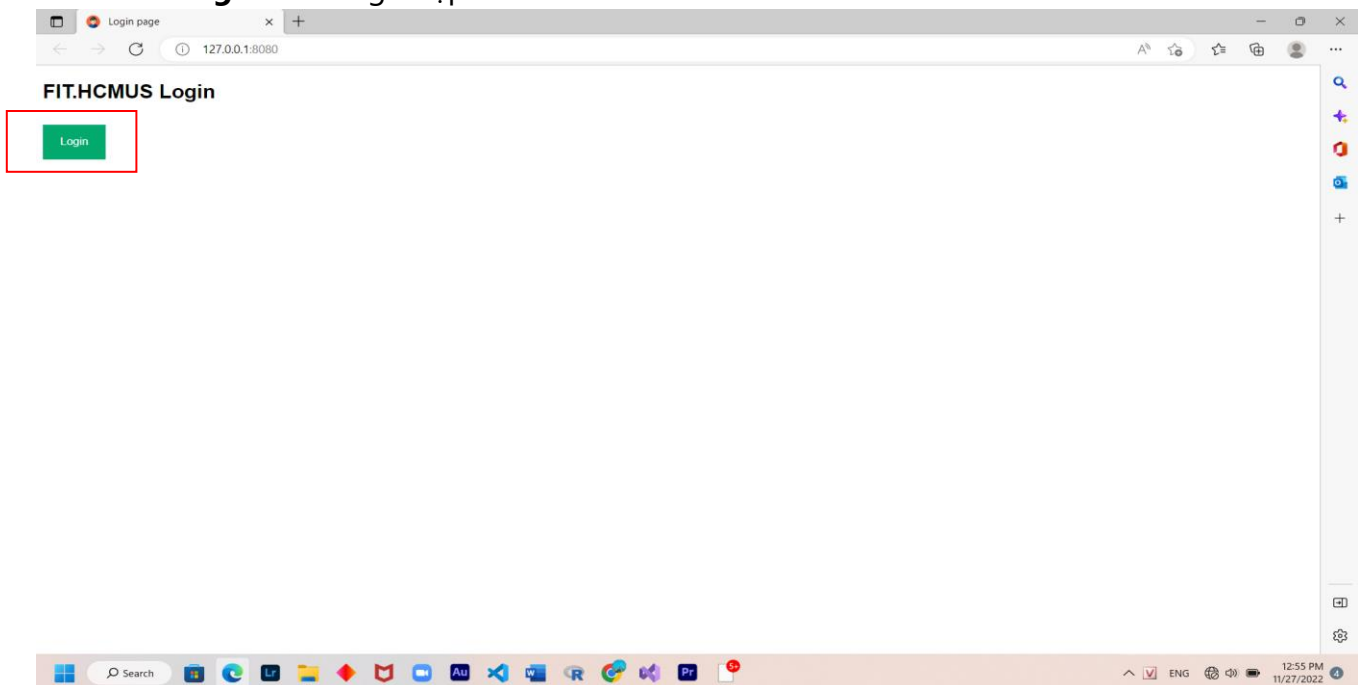
Còn nếu như bạn truy cập đến một trang web mà không thể tìm thấy máy chủ của website này thì bạn sẽ được dẫn tới website **404 NOT FOUND**



Bạn còn có thể mở nhiều trang website trên cùng một lần, đây được gọi là Multiple Connection. Khi bạn dừng chạy chương trình, một trang website bạn reload lại sẽ bị vắng nhưng những trang website còn lại vẫn sẽ không sao.

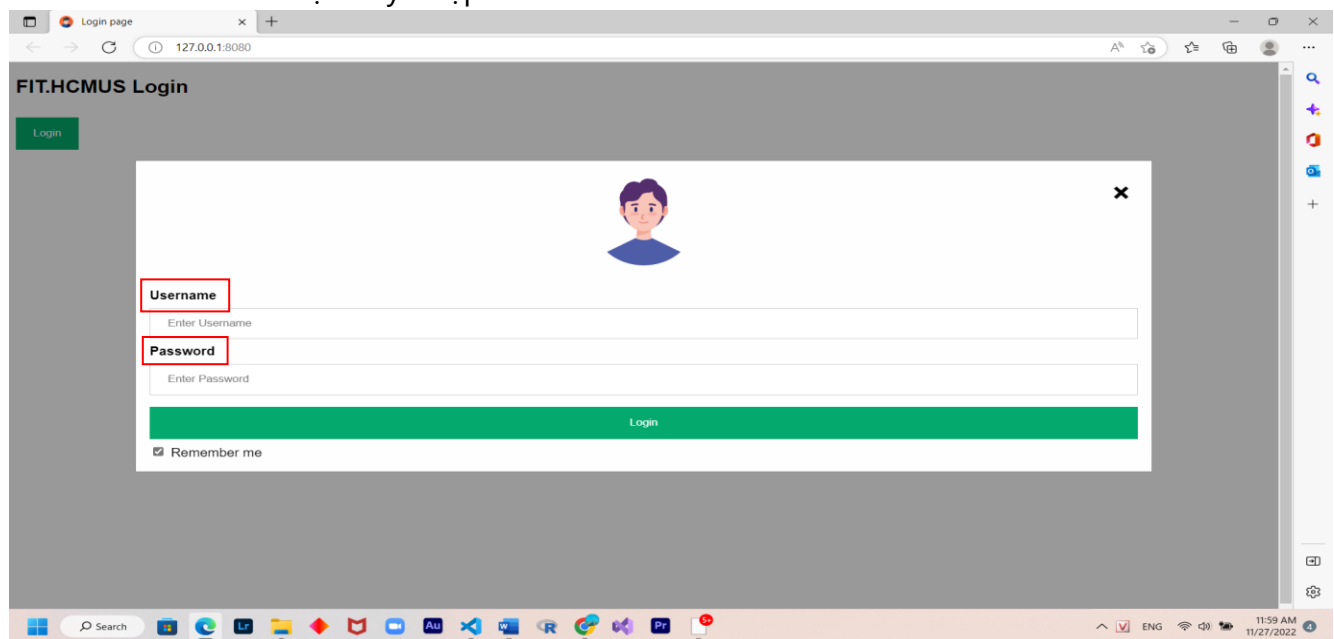


- **Bước 5:** Sau khi dẫn tới đường link website:
Ấn **Login** để đăng nhập

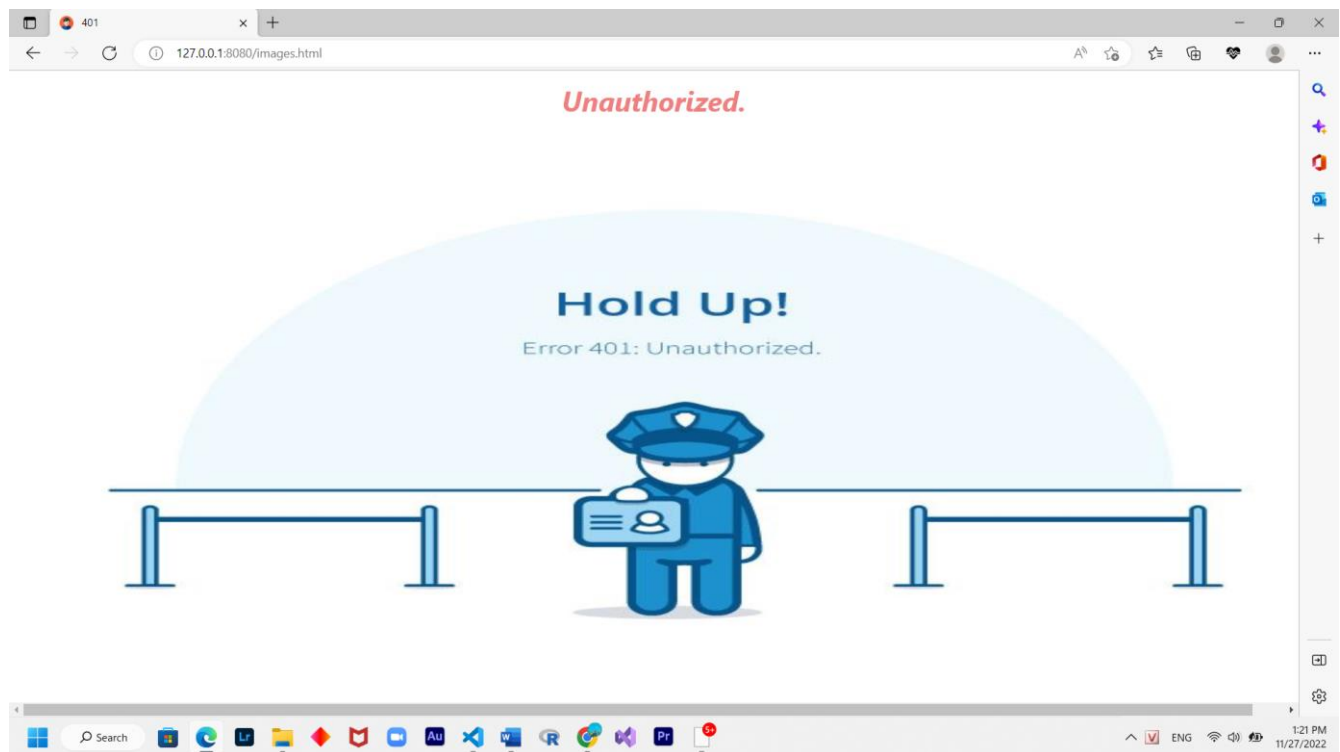


Ở **Username** bạn hãy nhập "admin"

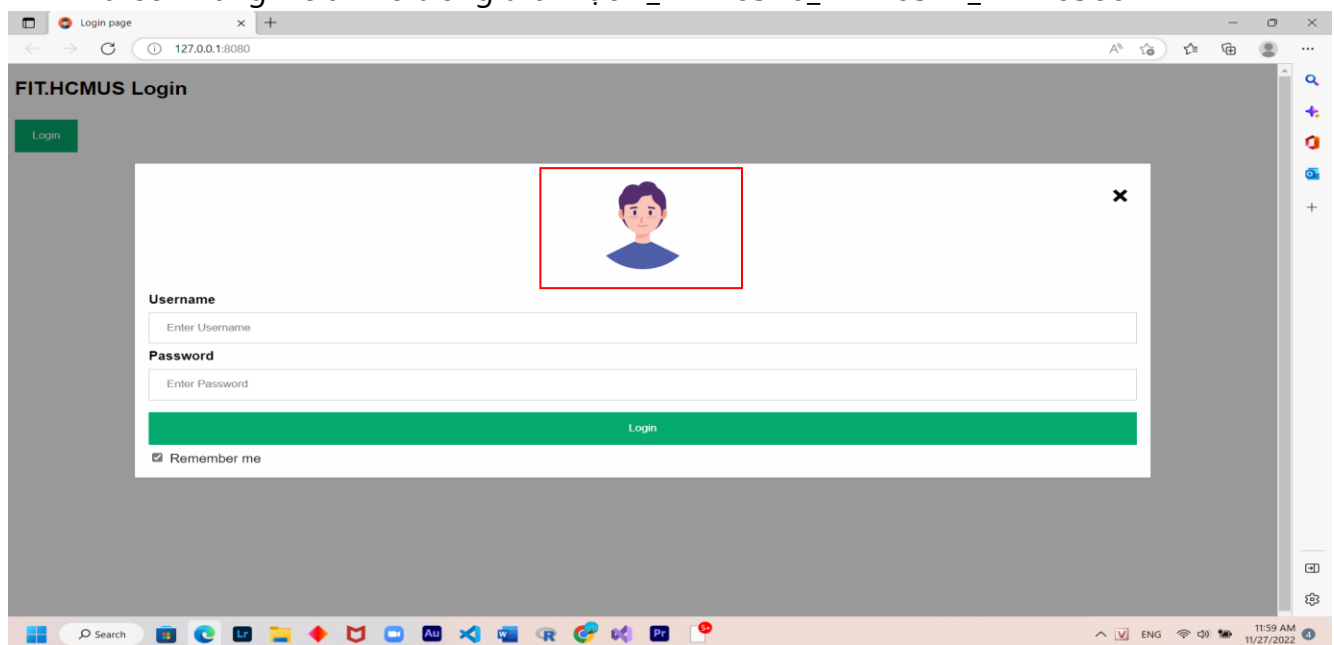
Ở **Password** bạn hãy nhập "123456"

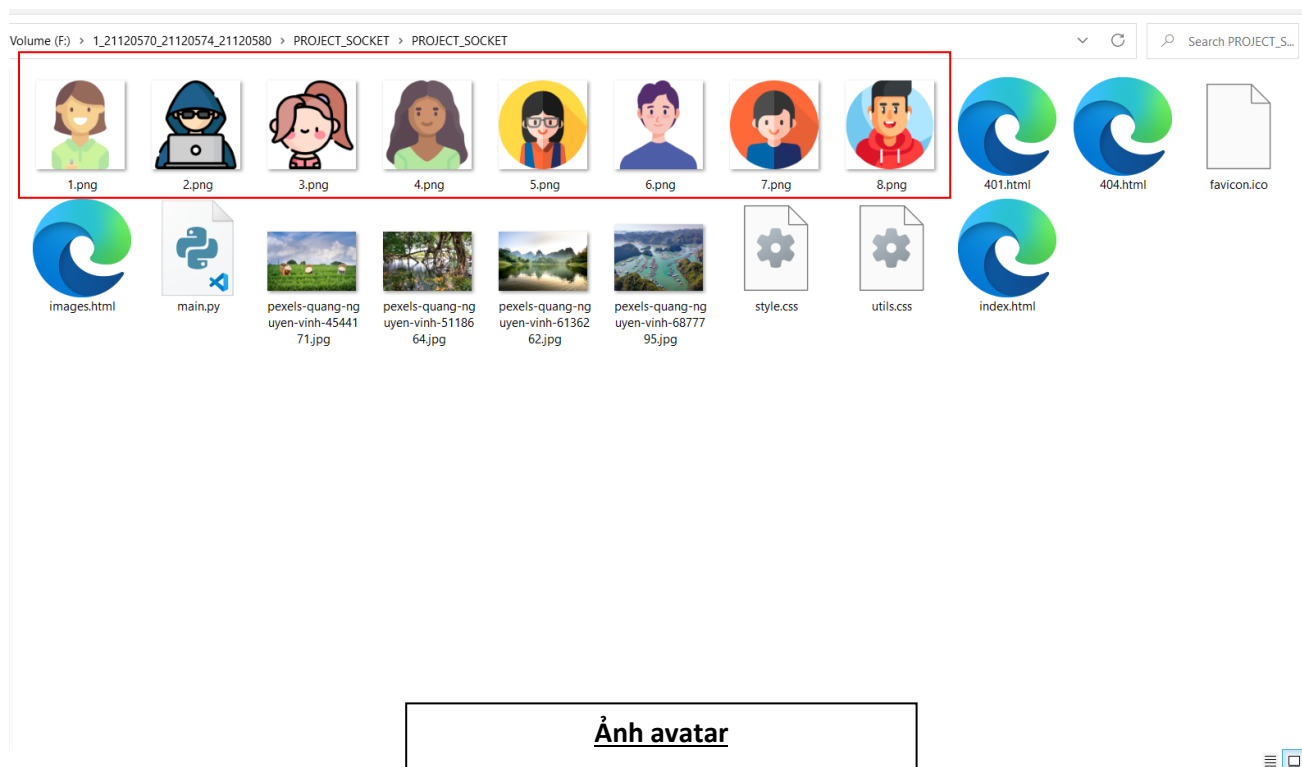


Nếu nhập sai Username hoặc Password thì bạn sẽ được dẫn tới đường dẫn website
401 Unauthorized

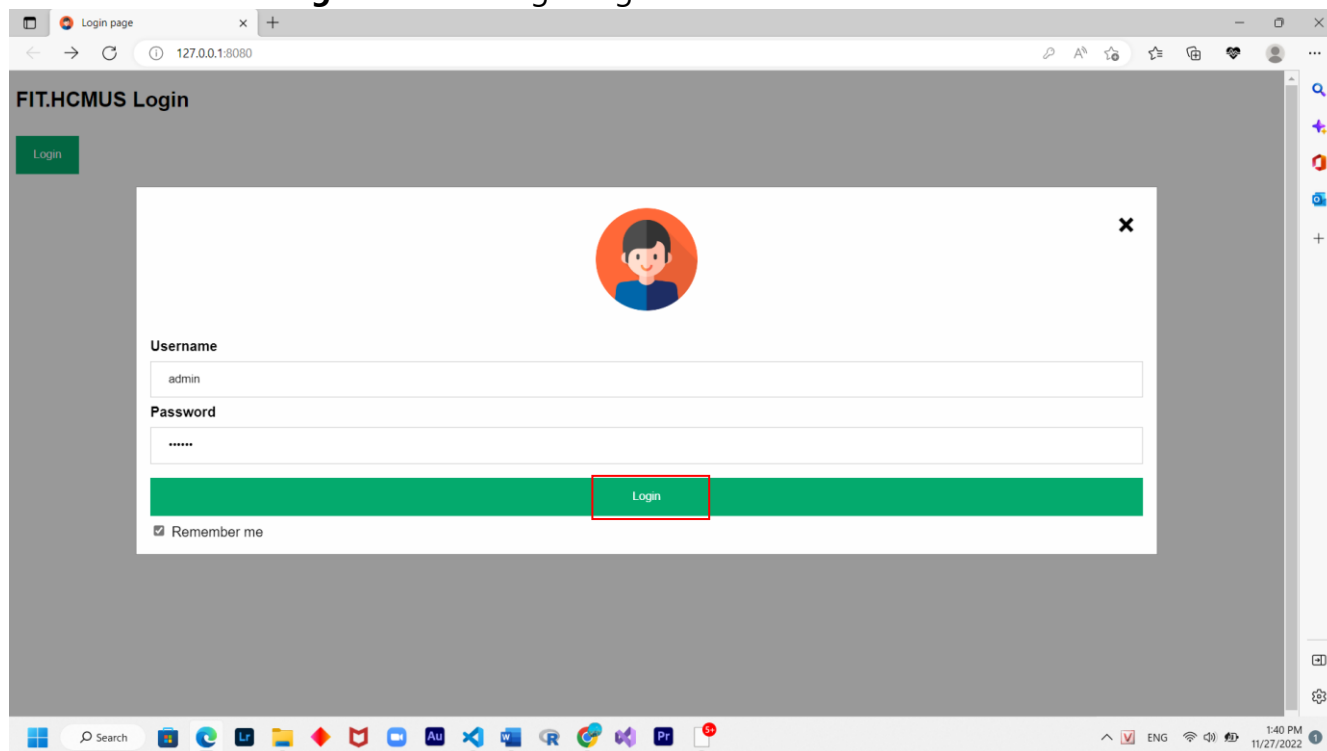


Và cứ mỗi lần bạn ấn vào **Login** để đăng nhập thì avatar tài khoản sẽ được thay đổi theo những file ảnh ở trong thư mục **1_21120570_21120574_21120580**

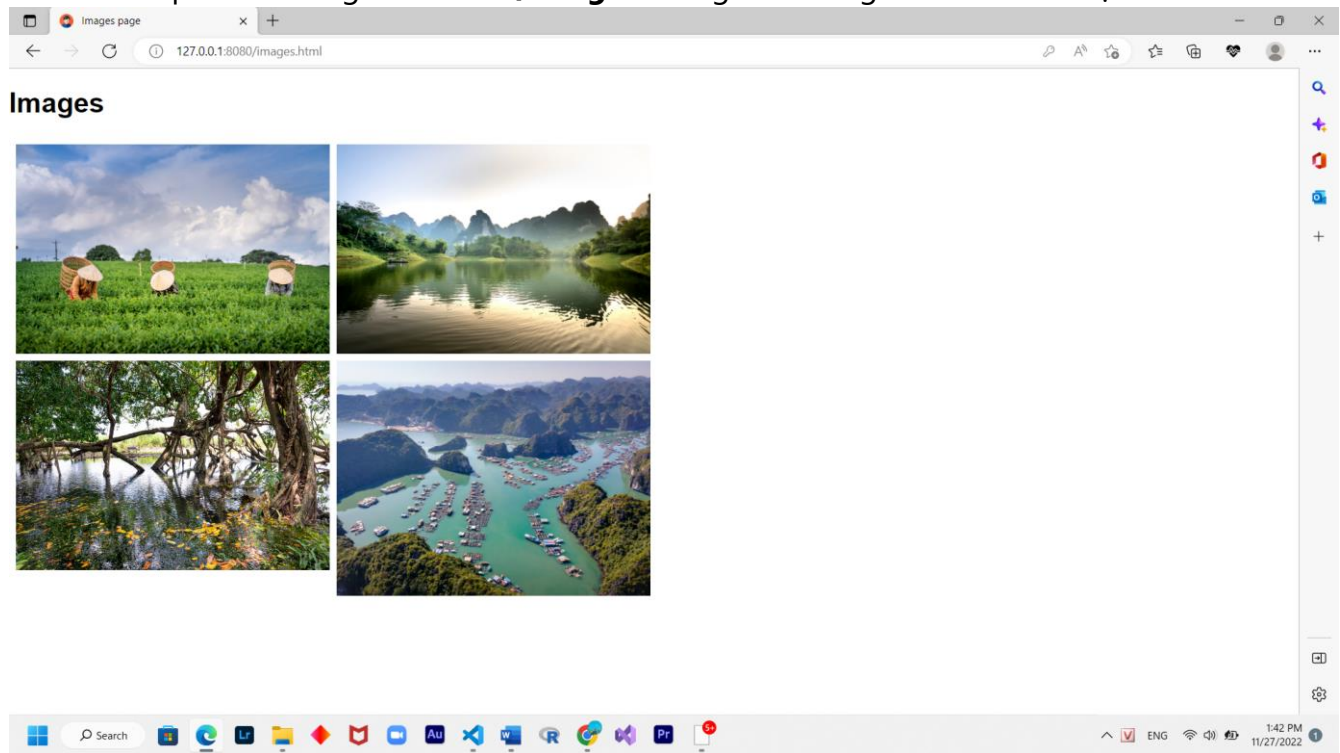




- **Bước 6: Ấn Login để vào trong trang web**



Kết quả cuối cùng: là dẫn tới **/image.html** gồm những hình ảnh đã chọn sẵn.



Và chỉ khi Client đóng kết nối thì Server mới dừng hẳn. Dấu hiệu để nhận biết là số data (dữ liệu) nhận được từ Client sẽ bằng 0.

BẢNG PHÂN CÔNG CÔNG VIỆC

STT	Họ và tên	MSSV	Công việc
1	Đặng Nguyễn Thanh Tín	21120570	Thực hiện phần "Multiple connection" Thực hiện phần "Lỗi page" Thực hiện phần "Multiple request" Thực hiện phần "Report"
2	Nguyễn Minh Trí	21120574	Thực hiện phần "Kết nối" Thực hiện phần "Đăng nhập" Thực hiện phần "Quản lý kết nối" Thực hiện phần "Đọc header"
3	Trần Thị Kim Trinh	21120580	Thực hiện phần "Tải được page index.html" Tổng hợp các code từ các thành viên Chạy thử chương trình

CÁC NGUỒN TÀI LIỆU THAM KHẢO

- 1) M.V.Cường, Giáo trình Mạng máy tính, Thành phố Hồ Chí Minh: Nhà xuất bản Đại học Quốc gia TP.HCM
- 2) Socket Programming in Python[Online]. Available: <https://realpython.com/python-sockets/>
- 3) What is web socket and how it is different from the HTTP?[Online]. Available: <https://www.geeksforgeeks.org/what-is-web-socket-and-how-it-is-different-from-the-http/>
- 4) A Simple Explanation Of What A WebSocket Is[Online]. Available: <https://www.wallarm.com/what/a-simple-explanation-of-what-a-websocket-is>
- 5) How do TCP/IP and HTTP work together? [Online]. Available: <https://stackoverflow.com/questions/46794420/how-do-tcp-ip-and-http-work-together>
- 6) How do multiple clients connect simultaneously to one port, say 80, on a server?[Online]. Available: <https://stackoverflow.com/questions/3329641/how-do-multiple-clients-connect-simultaneously-to-one-port-say-80-on-a-server>
- 7) Multiple Connections Server[Online]. Available: <https://es7evam.gitbook.io/security-studies/exploitation/sockets/02-multiple-connections-server>
- 8) Using multiple nodes[Online]. Available: <https://socket.io/docs/v4/using-multiple-nodes/>