

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

---



# **BÁO CÁO ĐỒ ÁN SOCKET**

**Mạng máy tính**

## **ĐỀ 1: WEB SERVER**

<b>Thông tin nhóm:</b>	<b>Trần Nguyên Huân – 21127050</b> <b>Lê Huỳnh Phúc – 21127392</b> <b>Nguyễn Phú Trọng - 21127708</b>
<b>Lớp:</b>	<b>21CLC03</b>
<b>Giảng viên hướng dẫn:</b>	<b>Lê Hà Minh</b>

23/12/2022

# CONTENTS

---

## Table of Contents

---

1. Bảng phân công công việc, mức độ hoàn thành.....	3
2. Kịch bản giao tiếp của chương trình .....	3
2.1. Tạo Server .....	3
2.2. Nhận yêu cầu từ Client và phản hồi.....	3
2.3. Đóng Server .....	4
3. Hướng dẫn sử dụng các tính năng của chương trình.....	4
3.1. Chạy chương trình Python để tạo Server .....	4
3.2. Truy cập Server thông qua địa chỉ của Server.....	5
3.3. Tiến hành đăng nhập .....	6
3.4. Lỗi 404 File Not Found.....	7
3.5. Nhiều client kết nối cùng lúc.....	8
3.5. Nhiều Request trong kết nối .....	8
4. Môi trường lập trình và các framework để hỗ trợ thực thi ứng dụng....	8
5. Tài liệu tham khảo .....	9

## 1. Bảng phân công công việc, mức độ hoàn thành

Họ và tên	MSSV	Công việc	Mức độ hoàn thành
Trần Nguyên Huân	21127050	- Xử lý phương thức “GET” - Xử lý các HTTP request - Quản lý kết nối	100%
Lê Huỳnh Phúc	21127392	- Xử lý phương thức “POST” - Xử lý HTTP response - Xử lý lỗi page 404 (error 404)	100%
Nguyễn Phú Trọng	21127708	- Viết báo cáo - Xử lý multiple client - Xử lý lỗi page 401(error 401)	100%

## 2. Kịch bản giao tiếp của chương trình

### 2.1. Tạo Server

- Server được lập trình bằng socket, ngôn ngữ lập trình là Python.
- Host sẽ dựa trên địa chỉ của máy tính tạo Server.
- Port sẽ là 8080.

### 2.2. Nhận yêu cầu từ Client và phản hồi

- Client sẽ kết nối với Server thông qua địa chỉ IP của Server.
- Sau khi có Client kết nối tới, Server sẽ chấp nhận truy cập.
- Tiếp theo, Client sẽ gửi yêu cầu để Server thực hiện.
- Ở đồ án này, yêu cầu của Client sẽ được thực hiện bởi Server thông qua việc đăng nhập tài khoản.

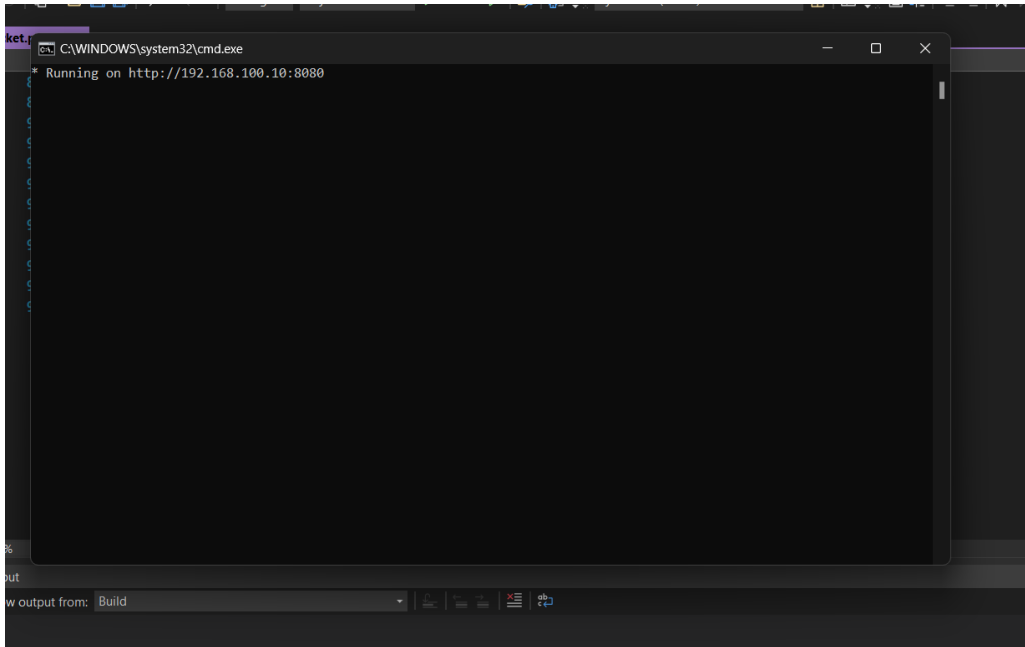
- + Nếu yêu cầu bắt đầu bằng “GET” và theo sau đó là “/” hoặc tên filename (index.html, css/style.css,...), cuối cùng là HTTP/1.1, thì Server sẽ phân tích thông tin yêu cầu của Client rồi load file tương ứng với yêu cầu đó.
- + Nếu file không tồn tại, sẽ trả về lỗi “404 File Not Found”.
- + Nếu yêu cầu bắt đầu là “POST”, thì Server sẽ nhận được thông tin “uname” và “psw” kèm theo trong body của yêu cầu. Sau đó, Server sẽ phân tích thông tin và kiểm tra nếu “uname” và “psw” chính xác thì trả về nội dung trang hình ảnh, nếu sai thì trả về lỗi “401 Unauthorized”
- Để có thể load file một cách chính xác thì cần chỉ định Content-type. Trong mỗi trang mà Server phản hồi yêu cầu từ Client sẽ có những loại file khác nhau, vì vậy, Content-type sẽ phụ thuộc vào loại file của file name.

### 2.3. Đóng Server

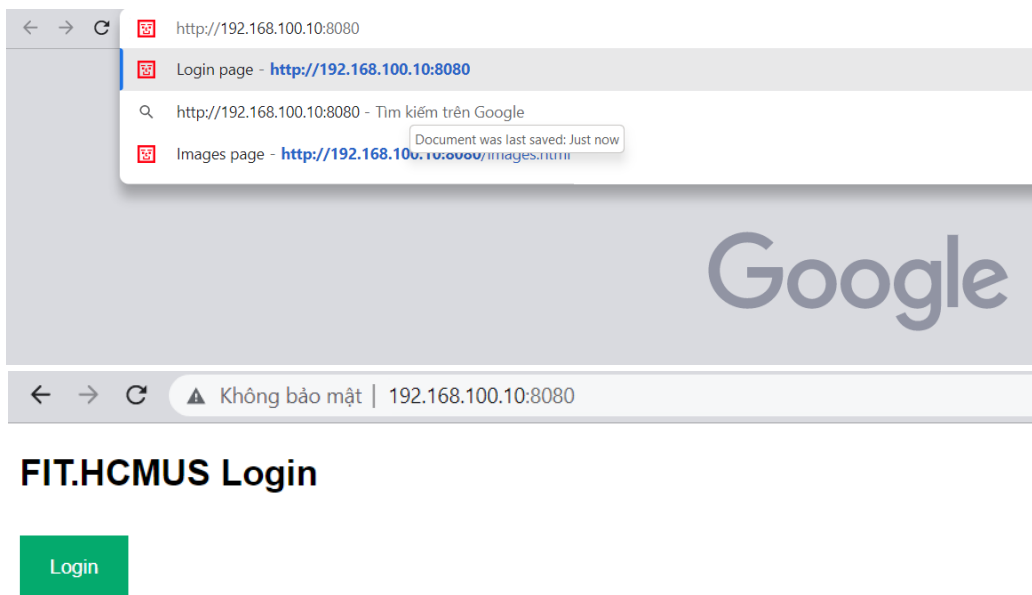
- Sau khi phân tích thông tin yêu cầu của Client để xác định filename, Server gửi phản hồi với HTTP header tương ứng, kèm theo body. Cuối cùng Server sẽ đóng kết nối.

## 3. Hướng dẫn sử dụng các tính năng của chương trình

### 3.1. Chạy chương trình Python để tạo Server

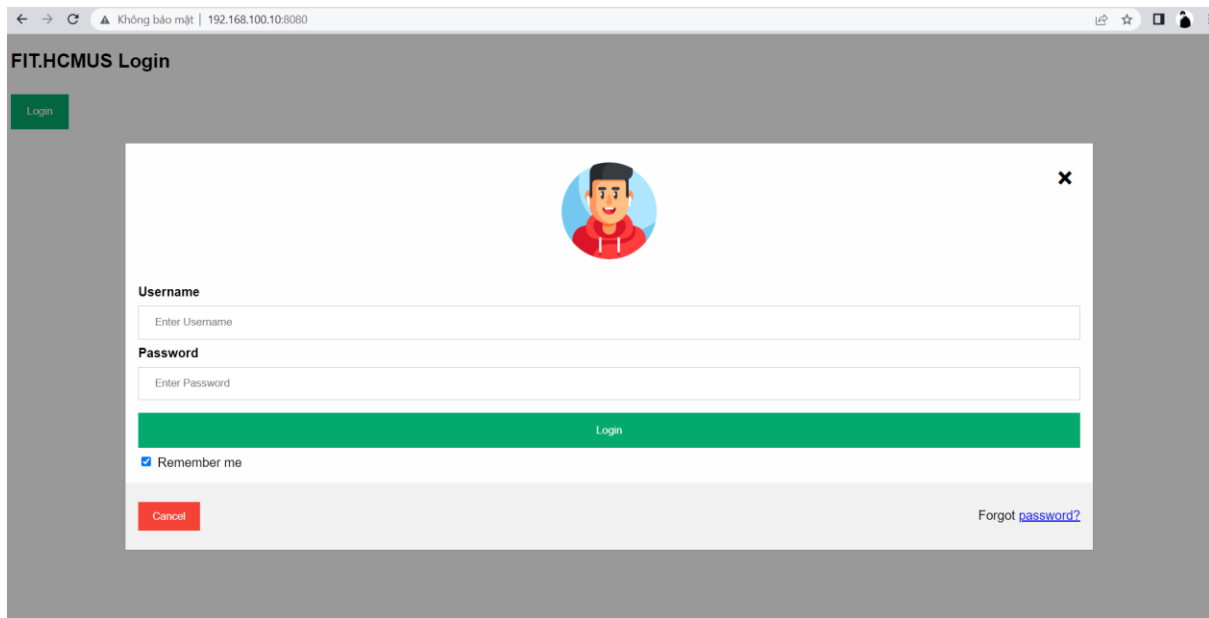


### 3.2. Truy cập Server thông qua địa chỉ của Server

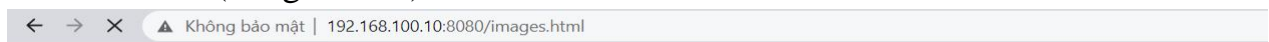


*Index.html*

### 3.3. Tiến hành đăng nhập



- Request bắt đầu với “POST” thông tin tài khoản và mật khẩu chính xác về server (uname = admin, psw = 123456) , sẽ trả về trang hình ảnh (images.html).

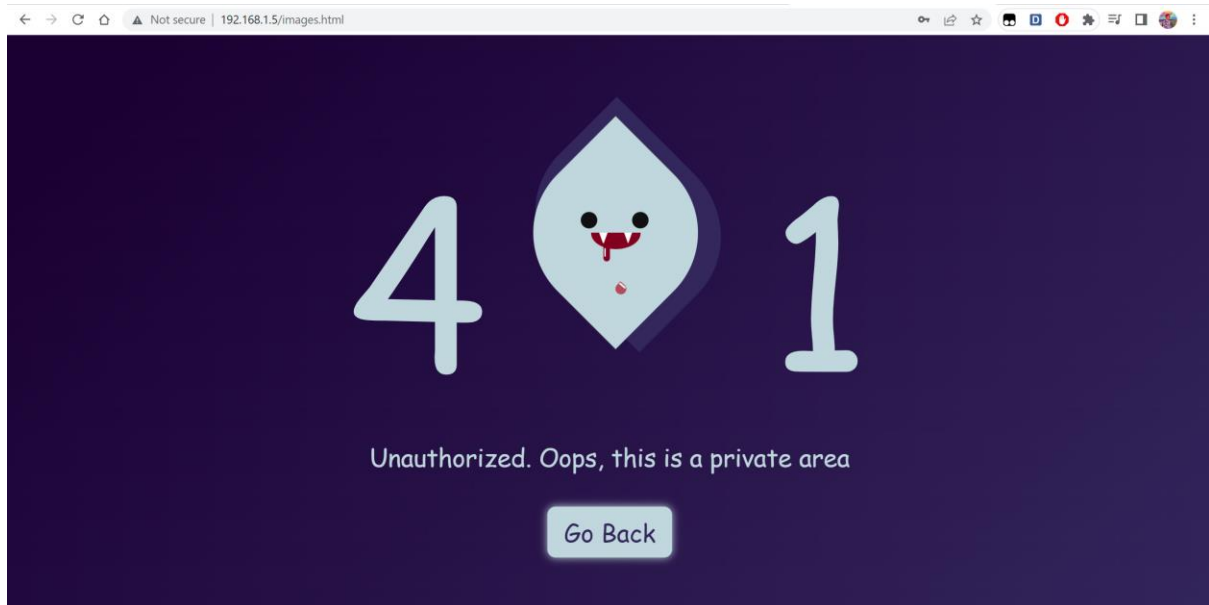


#### Images



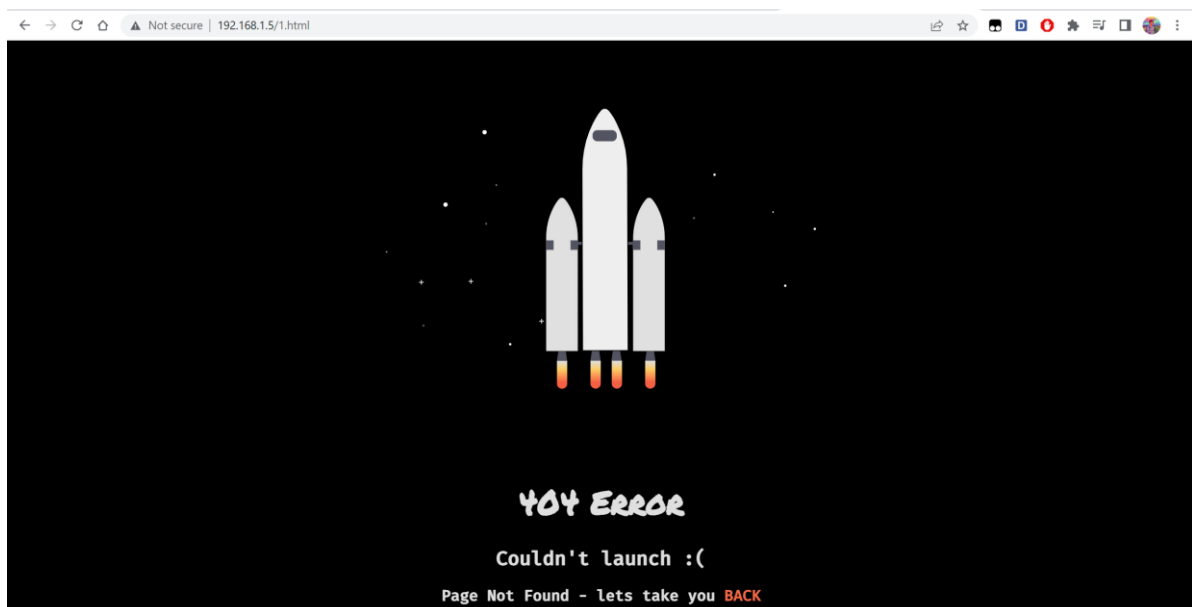
*Images.html*

- Nếu “POST” thông tin tài khoản hoặc mật khẩu về server không chính xác sẽ trả về lỗi “401 Unauthorized” (401.html)

*401.html*

### 3.4. Lỗi 404 File Not Found

- Trong quá trình server nhận request “GET” file nếu, file không tồn tại sẽ trả về lỗi “404 File Not Found” (404.html)

*404.html*

### 3.5. Nhiều client kết nối cùng lúc

```
* Running on http://192.168.1.5:8080
Client: ('192.168.1.5', 53377) da ket noi den server
<socket.socket fd=452, family=2, type=1, proto=0, laddr=('192.168.1.5', 8080), raddr=('192.168.1.5', 53377)>
GET / HTTP/1.1
Host: 192.168.1.5:8080
Connection: keep-alive
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/108.0.0.0 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9,vi;q=0.8
```

*Client 1*

```
Client: ('192.168.1.4', 60114) da ket noi den server
<socket.socket fd=476, family=2, type=1, proto=0, laddr=('192.168.1.5', 8080), raddr=('192.168.1.4', 60114)>
GET /css/style.css HTTP/1.1
Host: 192.168.1.5:8080
Connection: keep-alive
User-Agent: Mozilla/5.0 (Linux; Android 13; SM-G996B) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/108.0.0.0 Mobile Safari/537.36
Accept: text/css,*/*;q=0.1
Referer: http://192.168.1.5:8080/
Accept-Encoding: gzip, deflate
Accept-Language: vi-VN,vi;q=0.9,en-US;q=0.8,en;q=0.7,fr-FR;q=0.6,fr;q=0.5
```

*Client 2*

### 3.5. Nhiều Request trong kết nối

```
Client: ('192.168.1.5', 53555) da ket noi den server
<socket.socket fd=308, family=2, type=1, proto=0, laddr=('192.168.1.5', 8080), raddr=('192.168.1.5', 53555)>
GET / HTTP/1.1
Host: 192.168.1.5:8080
Connection: keep-alive
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/108.0.0.0 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9,vi;q=0.8

GET /css/style.css HTTP/1.1
Host: 192.168.1.5:8080
Connection: keep-alive
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/108.0.0.0 Safari/537.36
Accept: text/css,*/*;q=0.1
Referer: http://192.168.1.5:8080/
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9,vi;q=0.8

GET /css/style.css
GET /avatars/4.png HTTP/1.1
Host: 192.168.1.5:8080
Connection: keep-alive
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/108.0.0.0 Safari/537.36
Accept: image/avif,image/webp,image/apng,image/svg+xml,image/*,*/*;q=0.8
Referer: http://192.168.1.5:8080/
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9,vi;q=0.8

GET avatars/4.png
```

## 4. Môi trường lập trình và các framework để hỗ trợ thực thi ứng dụng

- Chương trình Web socket được lập trình trên Visual Studio 2022 bằng ngôn ngữ lập trình là Python.
- Framework hỗ trợ:
  - + Module socket trong python: Python cung cấp module socket giúp chúng ta dễ dàng thực hiện kết nối client server để giao tiếp với nhau



```
import socket
```

Tạo một socket với Socket Type là Sock\_Stream: TCP

```
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
```

+ Module threading trong python: cho phép chạy đa luồng với mục đích cho phép nhiều kết nối với server.

```
import threading
```

```
thread = threading.Thread(target=_handle, args=(conn, addr))
```

## 5. Tài liệu tham khảo

- Youtube:
  - + <https://www.youtube.com/watch?v=zAEEZydE1PY&t=809s>
  - + <https://www.youtube.com/watch?v=euMxTmYi1-o>
- Nguồn lấy file 401.html và 404.html: <https://freefrontend.com/html-css-404-page-templates/>
- Ngoài ra, còn có các file và bài giảng mà giảng viên cung cấp để thực hiện đồ án.