

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



# BÁO CÁO ĐỒ ÁN LẬP TRÌNH SOCKET

(ĐỀ 1: WEBSERVER)

*GV hướng dẫn: thầy Nguyễn Thanh Quân*

*Nhóm thực hiện:*

*21120542 – Lâm Hoàng Quốc*

*21120582 – Đinh Hoàng Trung*

*21120590 – Nguyễn Thủy Uyên*

## MỤC LỤC

<b>I. THÔNG TIN ĐỒ ÁN</b>	<b>2</b>
1. Bảng phân công công việc	2
2. Môi trường lập trình và các framework hỗ trợ	2
<b>II. KỊCH BẢN GIAO TIẾP CỦA CHƯƠNG TRÌNH</b>	<b>2</b>
1. Giao thức trao đổi giữa Client và Server	2
2. Cấu trúc thông điệp	3
a. Một Request nhận được:	3
b. HTTP Reponse.	5
3. Kiểu dữ liệu của thông điệp.	5
<b>III. HƯỚNG DẪN SỬ DỤNG TÍNH NĂNG</b>	<b>6</b>
1. Kết nối với Server	6
2. Đăng nhập.	6
<b>IV. ĐÁNH GIÁ MỨC ĐỘ HOÀN THÀNH ĐỒ ÁN</b>	<b>9</b>
<b>V. TÀI LIỆU THAM KHẢO</b>	<b>10</b>

**I. THÔNG TIN ĐỒ ÁN****1. Bảng phân công công việc**

MSSV	Tên Thành Viên	Nội Dung	Đánh Giá
21120542	Lâm Hoàng Quốc	-Kết nối. -Quản lý kết nối. -Tải được page login: parse request, đọc file, gửi file, handle request. -Multiple request.	96%
21120582	Đinh Hoàng Trung	-Tải được page login: Đọc file, response http header, handle request. -Multiple connection. -Multiple requests. -Report.	96%
21120590	Nguyễn Thủy Uyên	-Đăng nhập. - Multiple Connection. -Lỗi page. -Report.	100%

**2. Môi trường lập trình và các framework hỗ trợ**

- **Môi trường lập trình:** lập trình python trên Visual Studio Code. Với các thư viện hỗ trợ:
  - Thư viện os
  - Thư viện threading
  - Thư viện socket

**3. Ghi chú.**

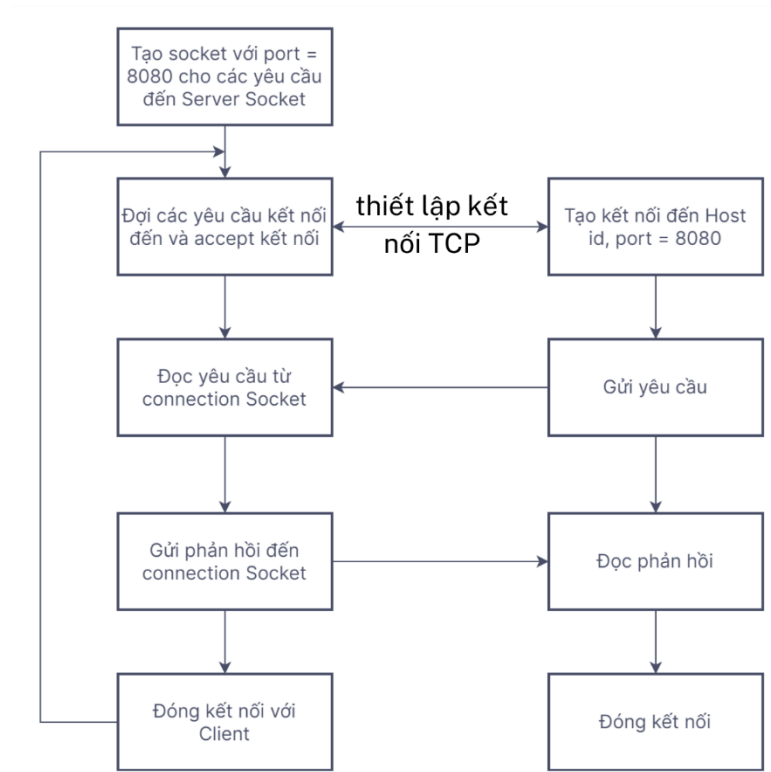
Ở đồ án này đối tượng giao tiếp của Server là Client cụ thể là trình duyệt web (web browser).

**II. KỊCH BẢN GIAO TIẾP CỦA CHƯƠNG TRÌNH****1. Giao thức trao đổi giữa Client và Server**

- **Giao thức trao đổi:**
  - Server tạo một socket với dạng địa chỉ Ipv4, giao thức TCP/IP, bind trên port 8080, sau đó listen và accept connection từ Client.
  - Server listen connection từ Client. Dợi connection từ Client nếu chưa nhận được tiếp tục listen connection từ Client.
  - Sau khi nhận được tín hiệu kết nối từ client: Accept kết nối từ Client.
  - Nhận dữ liệu từ Client, parse request và gửi dữ liệu đến Client tùy theo request đã phân tích từ dữ liệu nhận được.
  - Đóng kết nối.

## SERVER

## CLIENT



Kịch bản giao tiếp (hình 2.1).

- **Multiple Connection:**

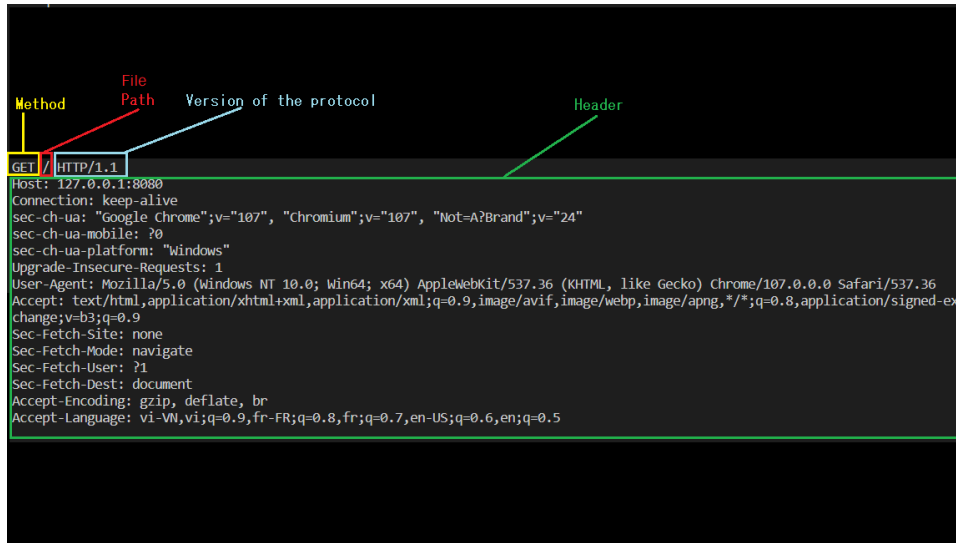
- Server tạo một socket để lắng nghe khi nào có kết nối mới từ client.
- Với mỗi một kết nối đến Server, sau khi chấp nhận kết nối, Server tạo ra một socket mới với một luồng mới để chờ nhận gói tin và phản hồi lại, đến khi nào không còn nhận data từ client thì đóng kết nối. Khi nhận được gói tin sẽ phân tích yêu cầu và gửi lại gói tin theo yêu cầu.

- **Multiple Request:**

- Client có thể gửi nhiều request đến server chỉ trong một connection.
- Server sẽ tiếp tục nhận và xử lý request cho đến khi bắt nhận về bằng 0 thì đóng kết nối.

## 2. Cấu trúc thông điệp

### a. Một Request nhận được:

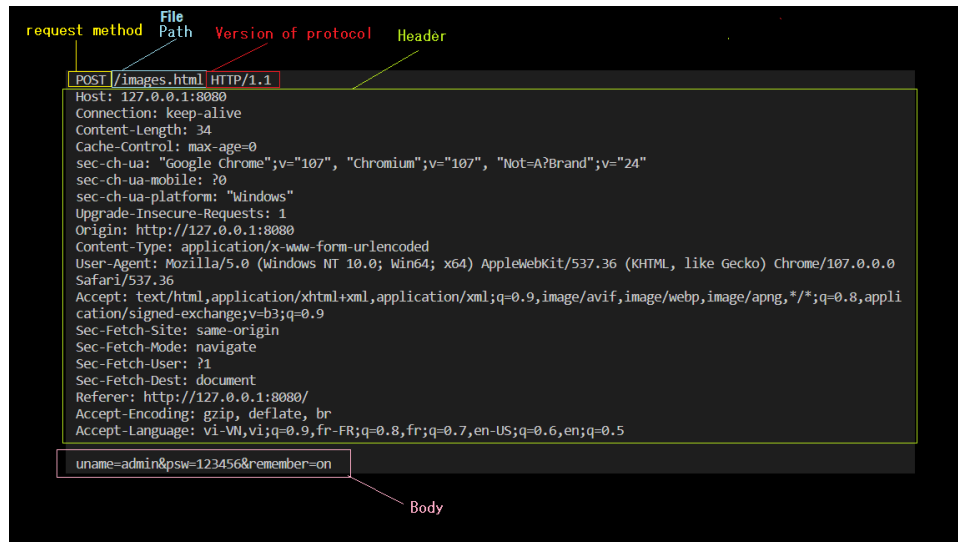


GET Request (hình 2.2)

- Với thông điệp yêu cầu, chúng ta cần xác định được phương thức, đường dẫn và phiên bản HTTP. Tiếp đó là thông tin của các header.
- Trong đồ án này ta cần quan tâm chủ yếu đến Request Line (dòng đầu tiên của request) gồm:
  - Request Method.
  - Path.
  - Version of the protocol.
 => Server sẽ parse từ request line lấy thông tin từ phần path để biết được cần load file nào cho request nếu là '/' thì sẽ được coi như là "index.html", tên file sẽ liên quan đến thư mục của máy chủ. Ứng với mỗi loại file server nhận được sẽ gửi reponse với http header tương ứng.

Filename	Content-Type:
*.html, *.htm	Content-Type: text/html
*.txt	Content-Type: text/plain
*.jpg, *.jpeg	Content-Type: image/jpeg
*.gif	Content-Type: image/gif
*.png	Content-Type: image/png
*.css	Content-Type: text/css
(unknown)	Content-Type: application/octet-stream

- Với thành phần trước dấu '/' đầu tiên là Method với 2 dạng GET và POST:
  - **GET:** yêu cầu từ client đến server cần phải load file cho request.
  - **POST:** Server nhận được thông tin đăng nhập. Cụ thể là thông tin của "uname" và "psw" tại phần body. (như hình 2.3)



Hình 2.3

### b. HTTP Reponse.

#### i. 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 Response.
- **404 – Not Found**
  - ⇒ Trả về khi máy khách truy cập một tài nguyên không tồn tại, server trả lời lại bằng status code 404 để thông báo rằng tài nguyên này không thể được đáp ứng.
- **401 – Unauthorized**
  - ⇒ Trả về khi máy khách đăng nhập tài khoản không hợp lệ, server trả lời lại bằng status code 401 để thông báo rằng đây là khu vực riêng tư bạn cần đăng nhập đúng tài khoản mới có thể vào được.

#### ii. Content type.

- Như đã nói trên tùy vào mỗi loại file gửi đi theo request nhận được thì reponse gửi về có http header với content type tương ứng:

<b>*.html, *.htm</b>	Content-Type: text/html
<b>*.txt</b>	Content-Type: text/plain
<b>*.jpg, *.jpeg</b>	Content-Type: image/jpeg
<b>*.gif</b>	Content-Type: image/gif
<b>*.png</b>	Content-Type: image/png
<b>*.css</b>	Content-Type: text/css
<b>(unknown)</b>	Content-Type: application/octet-stream

#### iii. Content Length:

Chiều dài file gửi đến client.

#### iv. Connection: keep-alive/close.

Trạng thái kết nối sau giao tiếp với client xong.

### 3. Kiểu dữ liệu của thông điệp.

Kiểu dữ liệu nhận và gửi đi đều dưới dạng nhị phân. Do đó muốn phân tích dữ liệu thì khi nhận về phải decode dữ liệu nhận về cũng như khi trước khi gửi data cho request ta cần thực hiện thao tác encode.

### III. HƯỚNG DẪN SỬ DỤNG TÍNH NĂNG

#### 1. Kết nối với Server

- Server chạy với địa chỉ IP 127.0.0.1, port là 8080. Client nhập địa chỉ IP của server với port là 8080 để connect với server.  
Ví dụ: http://127.0.0.1:8080.

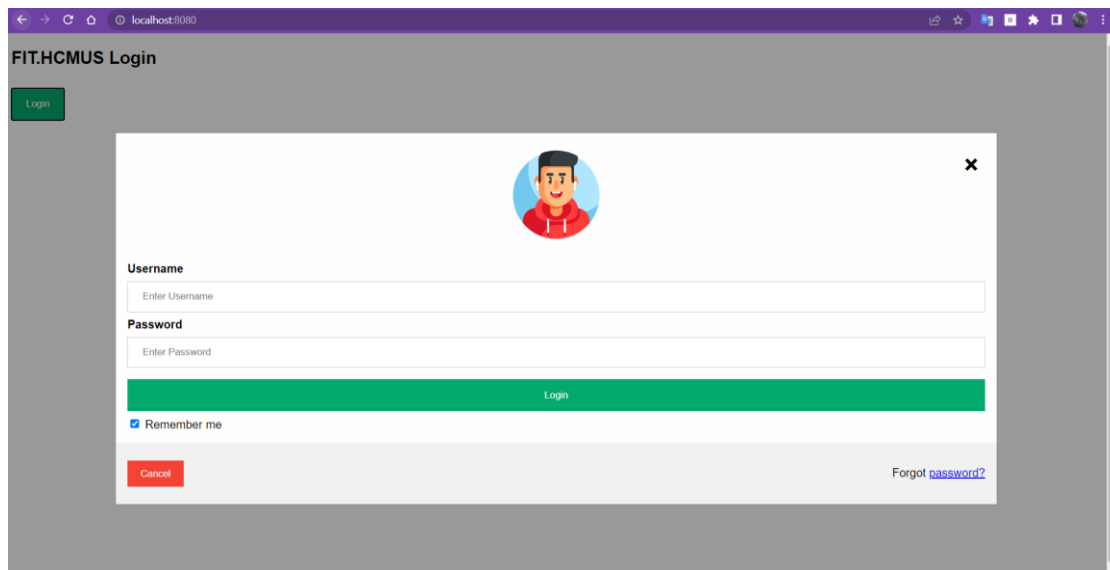
#### 2. Đăng nhập

- Sau khi truy cập và kết nối đến Server, Nếu kết nối thành công, màn hình Client sẽ hiện lên giao diện có nút đăng nhập (như hình 3.1).



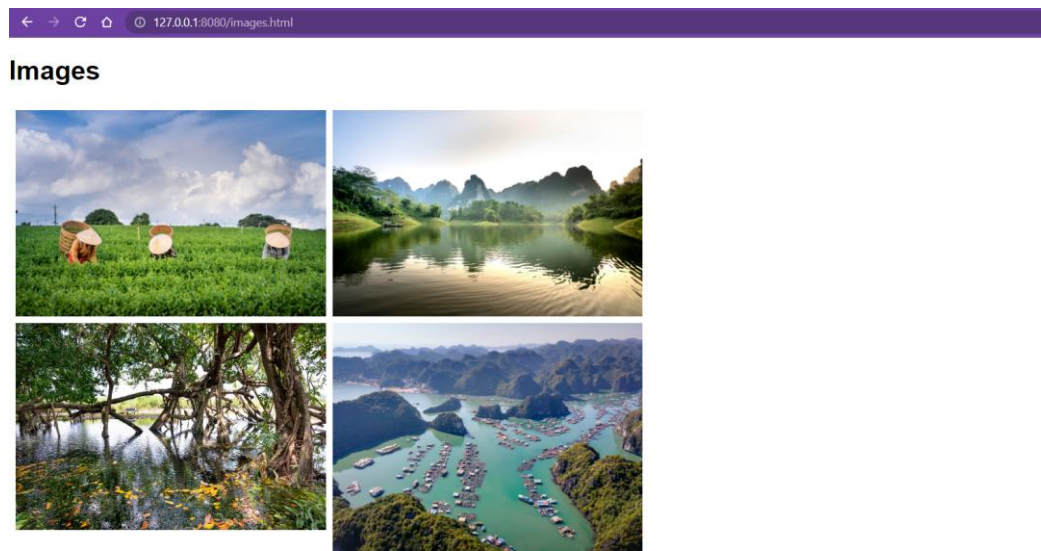
Hình 3.1

- Nhấn vào nút đăng nhập để đăng nhập: (như hình 3.2)



Hình 3.2

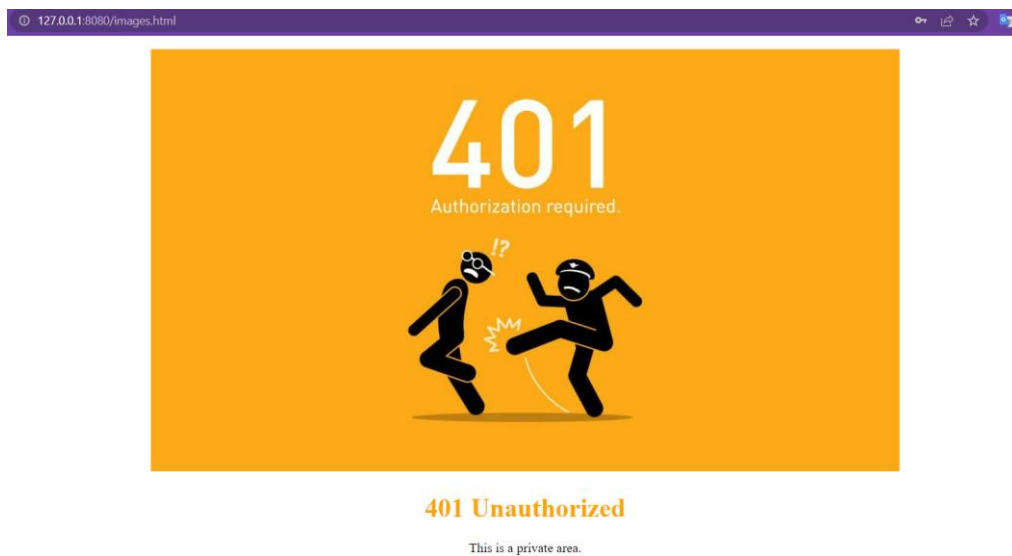
- Với một tài khoản đúng duy nhất là:
  - Username: admin.
  - Password: 123456.
- Nếu đăng nhập vào đúng tài khoản:
  - Client sẽ chuyển sang trang images với bốn bức hình. (như hình 3.3)



Hình 3.3

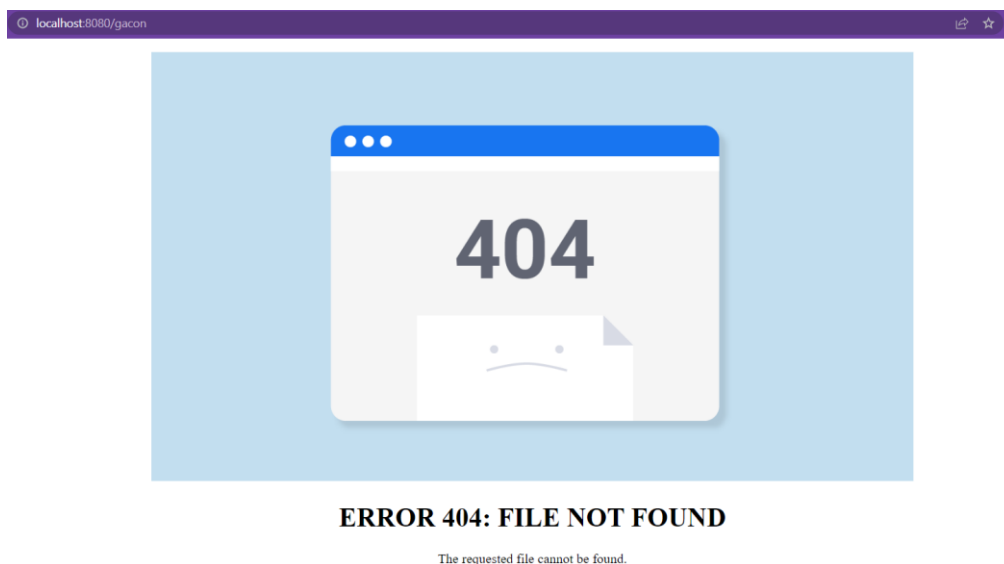


- Ngược lại nếu đăng nhập với tài khoản không hợp lệ Client chuyển sang trang 401 Unauthorized. (như hình 3.4)



Hình 3.4

- Lỗi load Page không đúng:
  - Client sẽ hiện lên lỗi 404 not found nếu địa chỉ truy cập đến Server với request là một file không tồn tại. (như hình 3.5)



Hình 3.5



Sơ đồ định tuyến (Hình 3.6)

#### IV. ĐÁNH GIÁ MỨC ĐỘ HOÀN THÀNH ĐỒ ÁN

	Chức năng	Đánh giá theo yêu cầu	Đánh giá mức độ hoàn thiện.
1	KẾT NỐI	Kết nối được	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	<p>Web Browser có thể render lên đầy đủ nội dung của trang index.html.</p> <p>Trang index.html hiển thị 1 HTML form cho phép đăng nhập (yêu cầu 4). Form này có phần action trỏ đến Web server.</p> <p>Chạy được các test case:</p> <p><a href="http://localhost:8080">http://localhost:8080</a></p> <p><a href="http://localhost:8080/index.html">http://localhost:8080/index.html</a></p> <p><code>http://&lt;ip&gt;:8080</code></p> <p><code>http://&lt;ip&gt;:8080/index.html</code></p>	100%

4	<b>ĐĂNG NHẬP</b>	<p><b>POST</b> method, gửi "uname" là "admin" và "psw" là "123456"</p> <p>Nếu đăng nhập thành công: chuyển đến images.html, hiển thị 4 hình ảnh như hình 3.3.</p> <p>Nếu đăng nhập không đúng tài khoản hiện trang lỗi đăng nhập không đúng (401 Unauthorized) như hình 3.4.</p>	100%
5	<b>Lỗi page</b>	<p>Load được 404 khi load page không đúng khi lỗi page (404 not found) (Không có file trong dữ liệu máy chủ).</p> <p>Chạy được các test case:</p> <p><a href="http://localhost:8080/gacon">http://localhost:8080/gacon</a></p> <p><a href="http://localhost:8080/fit.html">http://localhost:8080/fit.html</a></p> <p>http://&lt;ip&gt;:8080/gacon</p> <p>http://&lt;ip&gt;:8080/fit.html</p>	100%
6	<b>Multiple requests</b>	Gửi được nhiều requests trong một connection: chỉ đóng kết nối khi không còn nhận được data từ client (data nhận được bằng 0). Chưa biết khi nào Web Client gửi "Connection: close" để đóng kết nối.	90%
7	<b>Multiple connection</b>	Concurrent, handle được nhiều client cùng một lúc: theo xử lý đa luồng song song.	100%
8	<b>Report</b>	Báo cáo theo đúng qui định (đáp ứng những yêu cầu cơ bản, không chứa source code).	100%

## V. TÀI LIỆU THAM KHẢO

### Xây Dựng Webserver (Youtube):

- [https://www.youtube.com/watch?v=ryIGwYvPUUk&t=713s&ab\\_channel=L%C3%AANGuy%E1%BB%85n](https://www.youtube.com/watch?v=ryIGwYvPUUk&t=713s&ab_channel=L%C3%AANGuy%E1%BB%85n)
- [https://www.youtube.com/watch?v=zAEEZydE1PY&t=349s&ab\\_channel=L%C3%AANGuy%E1%BB%85n](https://www.youtube.com/watch?v=zAEEZydE1PY&t=349s&ab_channel=L%C3%AANGuy%E1%BB%85n)
- [https://www.youtube.com/watch?v=KZG66FcfZ1g&t=606s&ab\\_channel=L%C3%AANGuy%E1%BB%85n](https://www.youtube.com/watch?v=KZG66FcfZ1g&t=606s&ab_channel=L%C3%AANGuy%E1%BB%85n)

### Cấu Trúc Chương Trình Python:

- <https://cafedev.vn/tu-hoc-python-cau-truc-cua-mot-chuong-trinh-trong-python/>

Các Hàm String Python( để parse request):

- <https://toidicode.com/cac-ham-xu-ly-chuoi-trong-python-368.html>

Multithreading, xử lý đa luồng (đọc qua để làm multiconneciton):

- <https://viblo.asia/p/da-luong-trong-python-multithreading-WAyK8MO6ZxX>  
Tài liệu Giấy: Giáo trình “Mạng máy tính” Đại học Khoa học Tự nhiên bản 2020.