

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



BÁO CÁO ĐỒ ÁN THỰC HÀNH – LẬP TRÌNH SOCKET
MÔN MẠNG MÁY TÍNH

NHÓM THỰC HIỆN:

MSSV: 21120499 - HỌ TÊN: Nguyễn Duy Long

MSSV: 21120500 - HỌ TÊN: Mai Văn Minh

MSSV: 21120517 - HỌ TÊN: Nguyễn Thúc Minh Nhật

Giảng viên lý thuyết: Nguyễn Thanh Quân

Lớp lý thuyết: 21CTT5

Học kỳ - Niên khoá: HK1__2022-2023

Contents

I. THÔNG TIN THÀNH VIÊN..... 3

II. ĐÁNH GIÁ MỨC ĐỘ HOÀN THÀNH 3

III. KỊCH BẢN GIAO TIẾP CHƯƠNG TRÌNH 4

1. Giao thức trao đổi giữa Client và Server.....4

2. Cấu trúc thông điệp4

 a. Thông điệp yêu cầu 4

 b. Thông điệp phản hồi 5

3. Kiểu dữ liệu thông điệp6

4. Cách tổ chức cơ sở dữ liệu.....6

IV. MÔI TRƯỜNG LẬP TRÌNH VÀ CÁC FRAMEWORK HỖ TRỢ ĐỂ THỰC THI ỨNG DỤNG 6

V. HƯỚNG DẪN SỬ DỤNG CÁC TÍNH NĂNG CHƯƠNG TRÌNH..... 6

1. Phía Server.....6

2. Phía Client.....9

VI. BẢNG PHÂN CÔNG CÔNG VIỆC TRONG ĐỒ ÁN..... 12

VII. TÀI LIỆU THAM KHẢO..... 13

THÔNG TIN CHUNG VỀ ĐỒ ÁN

I. THÔNG TIN THÀNH VIÊN

MÃ SỐ SINH VIÊN	HỌ VÀ TÊN
21120499	Nguyễn Duy Long
21120500	Mai Văn Minh
21120517	Nguyễn Thúc Minh Nhật

II. ĐÁNH GIÁ MỨC ĐỘ HOÀN THÀNH

CHỨC NĂNG	MỨC ĐỘ HOÀN THÀNH	MỞ RỘNG
Kết nối	100%	Không
Quản lý kết nối	100%	Không
Tải được page index.html	100%	Không
Đăng nhập	100%	Có
Lỗi page	100%	Không
Multiple Requests	100%	Không
Multiple Connection	100%	Không
Report	100%	Không

NỘI DUNG ĐỒ ÁN

III. KỊCH BẢN GIAO TIẾP CHƯƠNG TRÌNH

1. Giao thức trao đổi giữa Client và Server

- Giao thức trao đổi được sử dụng là giao thức TCP/IP ở tầng Transport.
- Giao thức trao đổi dữ liệu giữa client và server tuân thủ theo giao thức TCP, có nghĩa là bao gồm hướng kết nối và bắt tay 3 bước trước khi truyền nhận dữ liệu của mỗi bên
- Một client muốn kết nối với server thì yêu cầu nó phải kết nối thông qua IP và Port của server, sau đó bên phía server xác nhận kết nối gửi lại gói tin và bắt đầu truyền nhận dữ liệu.
- Mỗi yêu cầu hay phản hồi từ client hay server được gọi là một thông điệp. Mỗi thông điệp này phải tuân thủ theo cấu trúc thông điệp mà giao thức yêu cầu, nếu như vi phạm cấu trúc thông điệp thì quá trình trao đổi dữ liệu giữa 2 bên sẽ xảy ra vấn đề.
- Ngay sau khi server chấp nhận kết nối và gửi lại gói tin để xác nhận việc truyền nhận dữ liệu, thì bên phía client gửi thông điệp yêu cầu(request) tới phía của server, server sẽ nhận request từ phía client rồi phân tích yêu cầu trong request đó. Nếu request hợp lệ, chứa đối tượng mà server có thể cung cấp thì server có thể gửi đối tượng đó cho client, còn nếu thông tin trong thông điệp yêu cầu sai thì server gửi lại thông điệp phản hồi dưới dạng một thông báo không chứa đối tượng. Thông điệp phản hồi của server được gọi là response.
- Ngay sau khi việc truyền nhận dữ liệu kết thúc thì bên phía client có thể đóng kết nối chủ động(client lúc này đã nhận đủ file mong muốn), lúc này bên phía server xác nhận việc client đã đóng kết nối bằng cách nó không nhận được request của client nữa. Lúc này server cũng đóng kết nối với client này nhưng vẫn đợi kết nối hoặc giữ kết nối với những client khác.

2. Cấu trúc thông điệp

- Tùy thuộc vào thông điệp được gửi từ bên nào mà cấu trúc thông điệp sẽ có sự thay đổi. Có 2 dạng thông điệp là thông điệp yêu cầu (request – client) và thông điệp phản hồi (response – server).

a. Thông điệp yêu cầu

```
GET / HTTP/1.1
Host: localhost:8080
Connection: keep-alive
sec-ch-ua: "Microsoft Edge";v="107", "Chromium";v="107", "Not=A?Brand";v="24"
sec-ch-ua-mobile: ?0
sec-ch-ua-platform: "Windows"
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
```

Hình 1: Thông điệp yêu cầu

- Thông điệp yêu cầu trên được viết dưới dạng mã ASCII nên do đó ai có kiến

thức thông thường về máy tính đều có thể hiểu được

- Một thông điệp yêu cầu thì thông thường sẽ có số lượng dòng không cố định, mỗi dòng đều kết thúc bằng dấu hiệu xuống dòng và trả về đầu dòng \r\n. Dòng cuối cùng của thông điệp có thể một \r\n nữa.
- Dòng đầu của thông điệp yêu cầu là *dòng yêu cầu* [request line]; các dòng tiếp theo được gọi là *dòng tham số* [header line]. Dòng yêu cầu có 3 trường: Method(phương thức), URL(đường dẫn liên kết đến đối tượng yêu cầu) và version(phiên bản HTTP hiện tại).
- Trường Method có thể nhận một trong nhiều lựa chọn sau: GET, POST,... và trong bài này chỉ có 2 Method là GET và POST. GET được dùng khi duyệt yêu cầu đối tượng nào đó trong trường URL.
- Các dòng tham số có thể bao gồm: Host(thông tin của máy chủ chứa đối tượng được yêu cầu); Connection: close(hoặc keep-alive) – chứa thông tin trình duyệt báo cho máy chủ biết nó muốn dùng kết nối thường trực hay không thường trực; User-agent – cho biết loại trình duyệt đang gửi yêu cầu đến máy chủ.
- Theo sau các header line là phần thân thông điệp entity body, với phương thức GET thì có thể trống, nhưng với phương thức POST thì phần này dùng để chứa các tham số mà User đưa vào như username và password chẳng hạn

b. Thông điệp phản hồi

- Thông điệp phản hồi cũng được viết dưới dạng mã ASCII như thông điệp yêu cầu.

```
HTTP/1.1 200 OK
Date: Mon, 27 Jul 2009 12:28:53 GMT
Server: Apache/2.2.14 (Win32)
Last-Modified: Wed, 22 Jul 2009 19:15:56 GMT
Content-Length: 88
Content-Type: text/html
Connection: Closed
```

```
<html>
<body>
<h1>Hello, World!</h1>
</body>
</html>
```

Hình 2: Thông điệp phản hồi

- Thông điệp phản hồi thông thường gồm có 3 phần: *dòng trạng thái* [status line], *dòng mô tả* [header line], *thân thông điệp* [entity body].
- Trước hết, với dòng trạng thái có 3 trường: version(phiên bản HTTP hiện tại), status code(mã trạng thái), phrase(thông báo tình trạng phản hồi từ tiến trình chủ). Ở hình trên tương ứng sẽ là HTTP/1.1 200 OK. Ngoài ra còn có nhiều status code và phrase

khác như 301 Moved Permanently, 400 Bad Request, 401 Not Found,...

- Đối với phần mô tả thì bao gồm những thông tin quan trọng nhất là: Content-Length(xác định chính xác kích thước của đối tượng được gửi); Content-type(xác định loại đối tượng trong phần thân thông điệp) và Connection(cho biết tình trạng kết nối là thường trực hay không thường trực) nhận 2 giá trị là *close* hoặc *keep-alive*, nếu là *keep-alive* có thể có thêm phần Keep-Alive (cho biết các điều kiện để giữ kết nối trong HTTP 1.1).

3. Kiểu dữ liệu thông điệp

- Dữ liệu nhận vào và gửi đi đều thuộc kiểu dữ liệu String (chuỗi ký tự).
- Dữ liệu được mã hoá và giải mã kiểu UTF-8.

4. Cách tổ chức cơ sở dữ liệu

- Các file cần có như avatars, css, images, images.html, index.html, 401.html, 404.html được chứa trong một folder duy nhất là web_src.
- Ban đầu, sau khi thiết lập và chấp nhận kết nối thì phía server sẽ gửi file index.html cho client, client sẽ render lên nội dung trang web. Để render được đầy đủ nội dung trang web thì cần có các file được nhúng sẵn vào mã nguồn của index.html. Do đó client tiếp tục gửi request để yêu cầu phía server cung cấp các file còn thiếu.
- Khi có một yêu cầu của client đến một đối tượng nào đó thì server sẽ dựa vào URL nhận được rồi truy xuất đến folder web_src để lấy đối tượng đó và gửi lại cho client, nếu đối tượng được yêu cầu không có trong folder web_src này thì bên phía server sẽ gửi lại 404.html hoặc nếu thông tin đăng nhập sai thì gửi lại file 401.html

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

- Ứng dụng được viết và chạy trên hệ điều hành Windows 10 và 11.
- Các file dữ liệu để thực thi ứng dụng được lưu trong folder web_src.
- Ngôn ngữ lập trình: Python, kết hợp thêm các thư viện hỗ trợ trong quá trình lập trình cho Server như socket, threading, os.
- Hệ điều hành: Tương thích tốt với Window OS.
- Không sử dụng bất kì framework nào.

V. HƯỚNG DẪN SỬ DỤNG CÁC TÍNH NĂNG CHƯƠNG TRÌNH

1. Phía Server

- Khi mở Server lên sẽ có thông báo trên cửa sổ Command Prompt như sau:

```
*Server running on http://localhost:8080
```

Hình 3: Thông báo đợi kết nối từ client

- Lúc này Server thông báo rằng đang đợi kết nối từ client tại IP: 127.0.0.1 với PORT 8080

```
GET / HTTP/1.1
Host: localhost:8080
Connection: keep-alive
sec-ch-ua: "Microsoft Edge";v="107", "Chromium";v="107", "Not=A?Brand";v="24"
sec-ch-ua-mobile: ?0
sec-ch-ua-platform: "Windows"
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/107.0.0.0 Safari/537.36 Edg/107.0.1418.42
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Sec-Fetch-Site: none
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
```

Hình 4: Thông điệp yêu cầu page index.html

- Khi có một client kết nối và gửi request thành công tới server thì sẽ hiển thị request của client qua cửa sổ Command Prompt. Ban đầu, request sẽ là trang index.html, file css, file png.

```
GET /css/style.css HTTP/1.1
Host: localhost:8080
Connection: keep-alive
sec-ch-ua: "Microsoft Edge";v="107", "Chromium";v="107", "Not=A?Brand";v="24"
sec-ch-ua-mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/107.0.1418.42
```

Hình 5: Thông điệp yêu cầu file style.css

```
GET /avatars/7.png HTTP/1.1
Host: localhost:8080
Connection: keep-alive
sec-ch-ua: "Microsoft Edge";v="107", "Chromium";v="107", "Not=A?Brand";v="24"
sec-ch-ua-mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/107.0.1418.42
```

Hình 6: Thông điệp yêu cầu file 7.png

- Tiếp đó, nếu như người dùng nhập username và password để login thì sẽ có request mới yêu cầu trang images.html, lúc này method sử dụng là POST method, với thông tin quan trọng phần body là username và password.

```

POST /images.html HTTP/1.1
Host: localhost:8080
Connection: keep-alive
Content-Length: 34
Cache-Control: max-age=0
sec-ch-ua: "Microsoft Edge";v="107", "Chromium";v="107", "Not=A?Brand";v="24"
sec-ch-ua-mobile: ?0
sec-ch-ua-platform: "Windows"
Upgrade-Insecure-Requests: 1
Origin: http://localhost:8080
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/107.0.0.0 Safari/537.36 Edg/107.0.1418.42
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Referer: http://localhost:8080/
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9

uname=admin&psw=123456&remember=on

```

Hình 7: Thông điệp yêu cầu file images.html dùng POST method

- Nếu như thông tin uname và psw mà User cung cấp chính xác thì server sẽ gửi lại trang images.html, file utils.css và các file jpg trong thư mục images để browser có thể render đầy đủ nội dung của trang images.html

```

GET /css/utils.css HTTP/1.1
Host: localhost:8080
Connection: keep-alive
sec-ch-ua: "Microsoft Edge";v="107", "Chromium";v="107", "Not=A?Brand";v="24"
sec-ch-ua-mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/107.0.1418.42

```

Hình 8: Thông điệp yêu cầu file utils.css

```

GET /images/pexels-quang-nguyen-vinh-4544171.jpg HTTP/1.1
Host: localhost:8080
Connection: keep-alive
sec-ch-ua: "Microsoft Edge";v="107", "Chromium";v="107", "Not=A?Brand";v="24"
sec-ch-ua-mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/107.0.1418.42

```

Hình 9: Thông điệp yêu cầu file jpg

```

GET /images/pexels-quang-nguyen-vinh-6877795.jpg HTTP/1.1
Host: localhost:8080
Connection: keep-alive
sec-ch-ua: "Microsoft Edge";v="107", "Chromium";v="107", "Not=A?Brand";v="24"
sec-ch-ua-mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/107.0.1418.42

```

Hình 10: Thông điệp yêu cầu file jpg

```

GET /images/pexels-quang-nguyen-vinh-5118664.jpg HTTP/1.1
Host: localhost:8080
Connection: keep-alive
sec-ch-ua: "Microsoft Edge";v="107", "Chromium";v="107", "Not=A?Brand";v="24"
sec-ch-ua-mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/107.0.1418.42

```

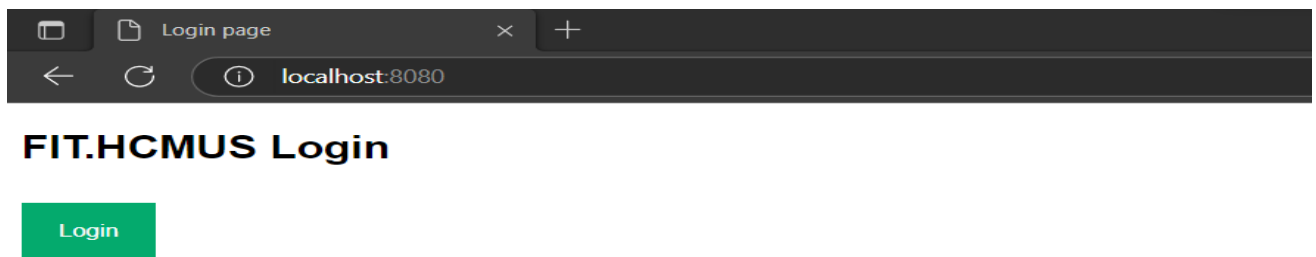
Hình 11: Thông điệp yêu cầu file jpg


```
GET /images/pexels-quang-nguyen-vinh-6136262.jpg HTTP/1.1
Host: localhost:8080
Connection: keep-alive
sec-ch-ua: "Microsoft Edge";v="107", "Chromium";v="107", "Not=A?Brand";v="24"
sec-ch-ua-mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
```

Hình 12: Thông điệp yêu cầu file jpg


2. Phía Client.

- Đầu tiên, để có thể trao đổi dữ liệu với Server, Client cần phải kết nối thành công tới IP và PORT tương ứng của Server, trong phần này Client cần truy cập đúng tới địa chỉ :
<http://localhost:8080/>
- Sau khi truy cập thành công, browser sẽ truy cập thành công được vào trang index.html với giao diện được cung cấp như sau.



Hình 13: Trang index.html

- Tiếp theo, để có thể truy cập được vào trang images.html thì User click tiếp vào ô **Login** góc phải màn hình và xuất hiện cửa sổ đăng nhập như sau:



Login

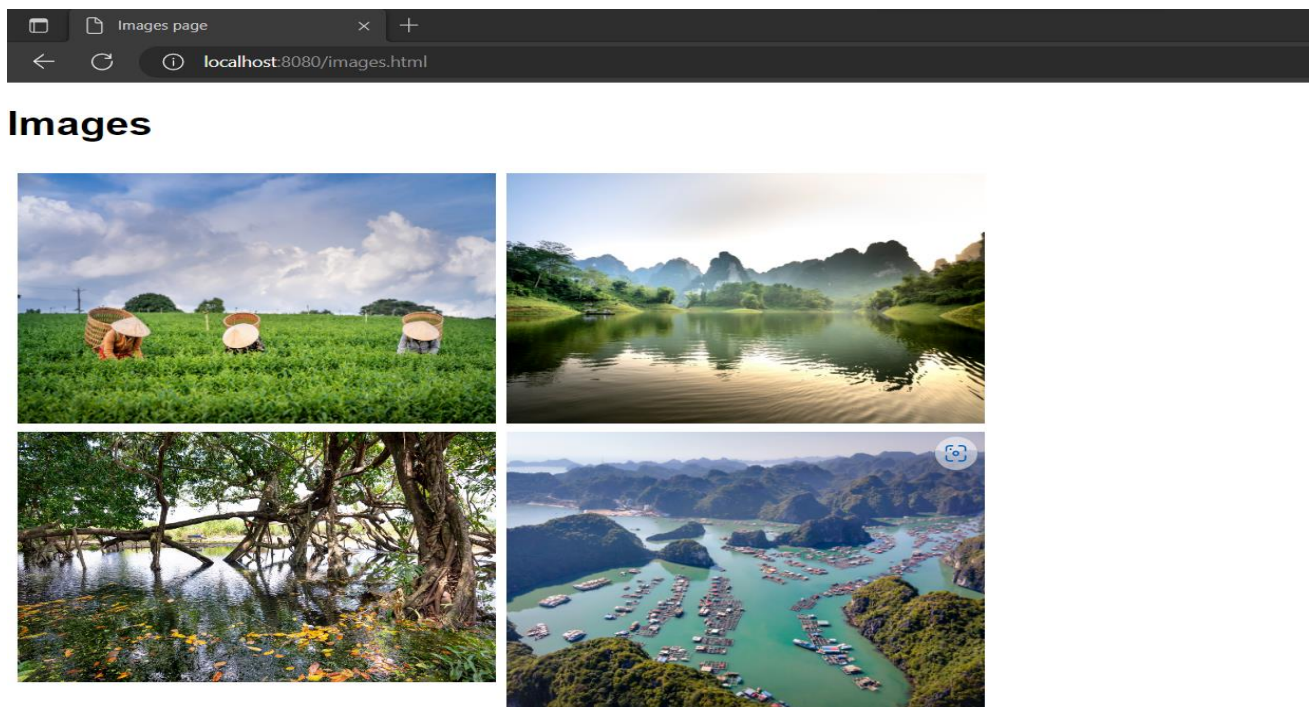
☒ Remember me

Cancel

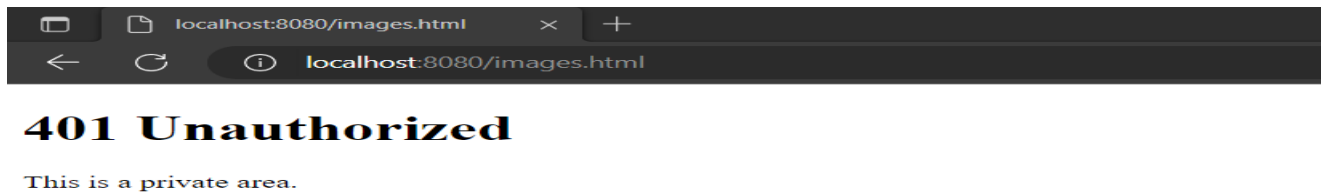
Forgot [password?](#)

Hình 14: Cửa sổ đăng nhập

- User nhập thông tin Username và Password tương ứng, nếu đăng nhập đúng thì User sẽ vào được trang images.html. Nếu thông tin đăng nhập sai thì User sẽ chuyển đến trang 401 Unauthorized.



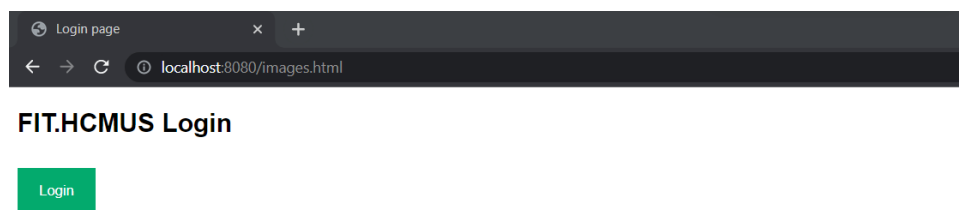
Hình 15: Trang images.html



Hình 16: Trang 401.html

- **Mở Rộng:** Nếu như User cố gắng truy cập trực tiếp vào page images.html mà chưa cung cấp uname và psw trước đó, thì phía Server sẽ đưa trở lại page index.html để yêu cầu cung cấp uname và psw.

```
*Server running on http://localhost:8080
Connected to ('127.0.0.1', 58185)
Request from ('127.0.0.1', 58185):
GET /images.html HTTP/1.1
Host: localhost:8080
Connection: keep-alive
```



VI. BẢNG PHÂN CÔNG CÔNG VIỆC TRONG ĐỒ ÁN

Bảng phân công công việc trong đồ án:

CHỨC NĂNG	THÀNH VIÊN	MỞ RỘNG	MỨC ĐỘ HOÀN THÀNH
Kết nối	Nguyễn Duy Long	Không	100%
Quản lý kết nối	Nguyễn Duy Long	Không	100%
Tải được page index.html	Nguyễn Thúc Minh Nhật	Không	100%
Đăng nhập	Mai Văn Minh	Gửi về trang đăng nhập (khi chưa đăng nhập) có tình truy cập vào image.html	100%
Lỗi page	Mai Văn Minh	Không	100%
Multiple Requests	Nguyễn Duy Long	Không	100%
Multiple Connection	Nguyễn Duy Long	Không	100%
Report	Mai Văn Minh	Không	100%

TÀI LIỆU THAM KHẢO

VII. TÀI LIỆU THAM KHẢO

- Các tài liệu thực hành được đăng tải trên Moodle của môn Mạng máy tính 21CTT5 – Khoa Công nghệ Thông tin – Trường Đại học Khoa học Tự nhiên – ĐHQG TP HCM.
- Tài liệu tham khảo Multiple Connection: <https://realpython.com/python-sockets/#multi-connection-server>
- Tài liệu xử lý TCP keep-alive: <https://tewarid.github.io/2013/08/16/handling-tcp-keep-alive.html>