

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN



ĐỒ ÁN SOCKET
MÔN: MẠNG MÁY TÍNH

LỚP 19CTT2

NHÓM THỰC HIỆN:

HỌ VÀ TÊN: TRỊNH THỊ THÙY
MSSV:19120390

HỌ VÀ TÊN: NGUYỄN ĐOÀN PHÚC
MSSV:19120330

A. PHÂN TÍCH ĐỒ ÁN

I. SOCKET - WEBSOCKET

1. Khái niệm Socket

Socket là một giao diện phần mềm trong các nút mạng đóng vai trò như một điểm cuối để gửi và nhận dữ liệu mạng. Một tiến trình gửi thông điệp đi và nhận các thông điệp về từ mạng thông qua Socket

2. WebSocket

Websocket là công cụ hỗ trợ việc kết nối qua lại trên internet giữa client và server. Giúp diễn ra nhanh chóng và hiệu quả hơn thông qua việc sử dụng TCP socket. Không chỉ sử dụng riêng cho ứng dụng web, Websocket có thể áp dụng cho bất kỳ ứng dụng nào khác cần có sự trao đổi thông tin trên Internet.

3. Ưu/nhược điểm của WebSocket

Ưu điểm: Websocket mang lại nhiều ưu điểm trong việc kết nối giữa client và server. Cụ thể như sau:

- Tăng tốc độ truyền tải thông tin giữa 2 chiều
- Dễ phát hiện và xử lý trong trường hợp có lỗi xảy ra
- Dễ dàng sử dụng, không cần cài đặt thêm các phần mềm bổ sung khác
- Không cần sử dụng nhiều phương pháp kết nối khác nhau

Nhược điểm: Một số nhược điểm của Websocket mà bạn cần lưu ý khi sử dụng có thể kể đến như:

- Chưa hỗ trợ trên tất cả các trình duyệt
- Với các dịch vụ có phạm vi yêu cầu, Websocket chưa hỗ trợ hoàn toàn.

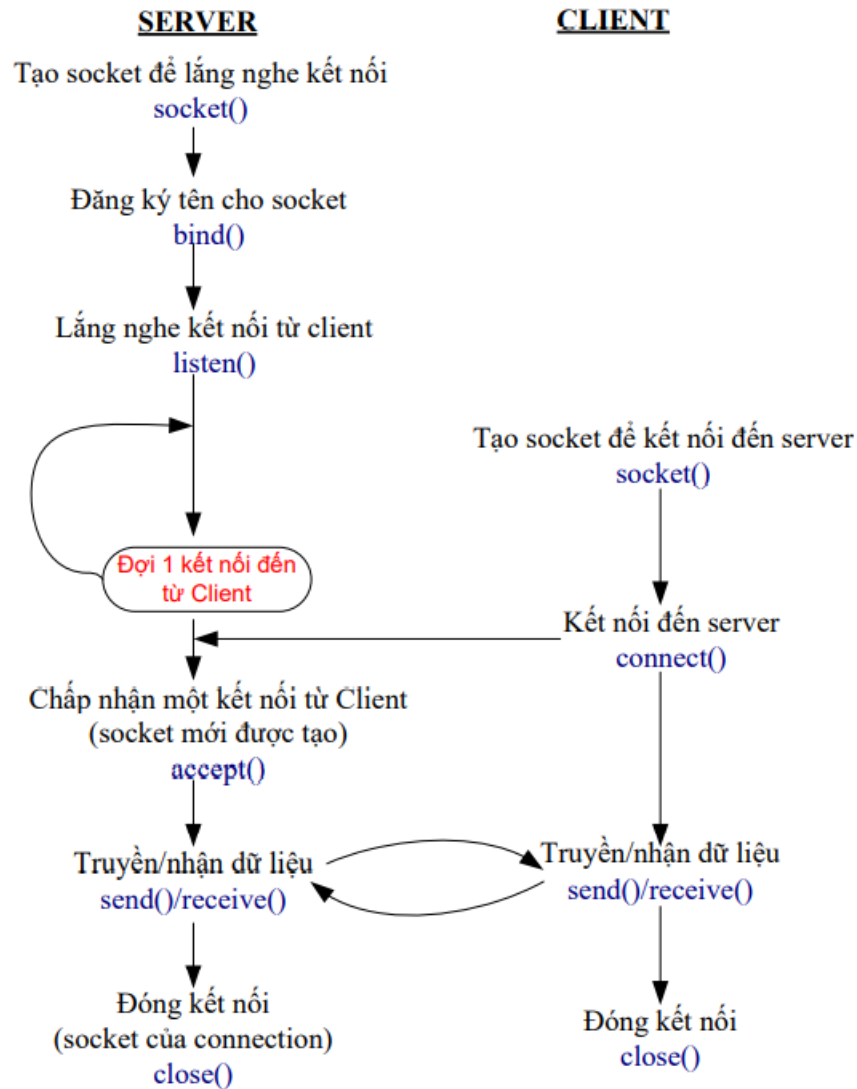
4. IP và Port

-Port xác định duy nhất một quá trình (process) trên một máy trong mạng.

Hay nói cách khác là cách mà phân biệt giữa các ứng dụng

-Một TCP/IP Socket gồm một địa chỉ IP kết hợp với một port. Xác định duy nhất một tiến trình (process) trên mạng. Hay nói cách khác luồng thông tin trên mạng dựa vào IP là để xác định máy một máy trên mạng còn port xác định 1 tiến trình trên 1 máy.

5. Hoạt động



(Sơ đồ tương tác giữa Server-client theo phương thức TCP)

II. GIAO THỨC TCP TRONG LẬP TRÌNH WEBSOCKET

Socket sử dụng giao thức TCP gọi là **Stream Socket**: Dựa trên giao thức TCP (Transmission Control Protocol) việc truyền dữ liệu chỉ thực hiện giữa 2 quá trình đã thiết lập kết nối. Giao thức này đảm bảo dữ liệu được truyền đến nơi nhận một cách đáng tin cậy, đúng thứ tự nhờ vào cơ chế quản lý luồng lưu thông trên mạng và cơ chế chống tắc nghẽn.

III. KIẾN TRÚC SERVER-CLIENT

1. Khái niệm

Mô hình **client-server** là một mô hình nổi tiếng trong mạng máy tính, được áp dụng rất rộng rãi và là mô hình của mọi trang web hiện có. Ý tưởng của mô hình này là máy con (đóng vai trò là máy khách) gửi một yêu cầu (request) để máy chủ (đóng vai trò người cung ứng dịch vụ), máy chủ sẽ xử lý và trả kết quả về cho máy khách.

2. Đặc điểm

-Server:

- Luôn luôn “sống”
- Chạy trên 1 địa chỉ cố định (IP tĩnh)
- Nhận và xử lý yêu cầu từ client

-Client:

- Liên lạc và gửi yêu cầu cho Server
- Có thể dùng IP “động”
- 2 client không thể liên lạc trực tiếp với nhau

3. Kiến trúc Server-Client và Socket

-Quy trình hoạt động của ứng dụng Server – Client như sau: Server có nhiệm vụ của là lắng nghe, chờ đợi kết nối từ Client trên địa chỉ IP của mình với PORT được quy định sẵn. Khi client gửi dữ liệu tới Server thì nó phải giải quyết một công việc là nhận dữ liệu đó -> xử lý -> trả kết quả lại cho Client.

-Client là ứng dụng được phục vụ, nó chỉ gửi truy vấn và chờ đợi kết quả từ server

IV. HTTP

1. Khái niệm

HTTP (HyperText Transfer Protocol) là giao thức truyền tải siêu văn bản được thiết kế và hoạt động theo kiểu Client-Server, sử dụng trong WWW dùng để truyền tải dữ liệu giữa WebServer đến các trình duyệt Web và ngược lại

2. Thông điệp HTTP

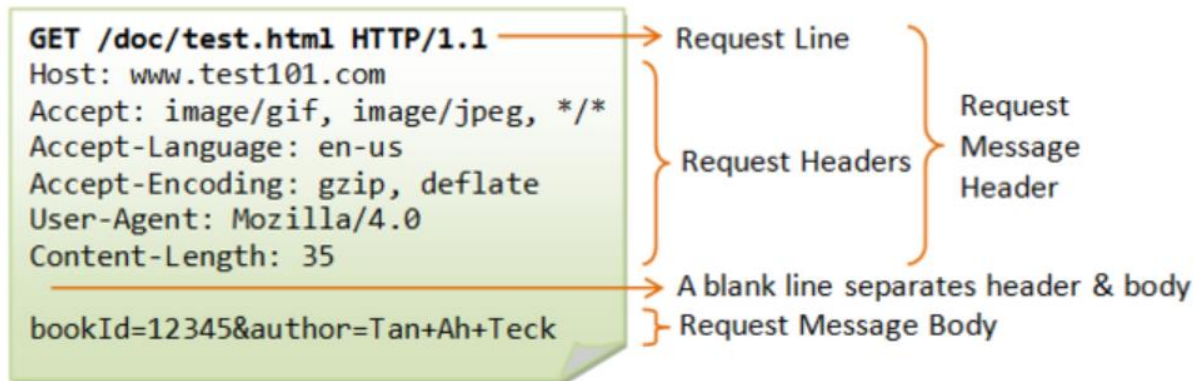
1. HTTP request

-Là thông tin Client gửi đến Server, yêu cầu Server thực hiện yêu cầu nào đó

-Cú pháp:

```
<method> <request-URL> <HTTP serverversion>  
<headers>  
<blank line>  
<body>
```

-Minh họa:



-Phân tích thành phần:

- Request-line:
 - method: có nhiều phương thức như GET, POST, HEAD, PUT, DELETE, OPTIONS,...
 - URL: địa chỉ định danh tài nguyên (đường dẫn). Trong trường hợp Request không yêu cầu tài nguyên cụ thể, URI có thể là dấu *
 - HTTP version là phiên bản HTTP đang sử dụng
- Headers: bao gồm Host, Accept, Accept-Language, Accept-Encoding, User-Agent, Content-Length,...
- Blank line: báo hiệu kết thúc Headers
- Body: có thể có hoặc không

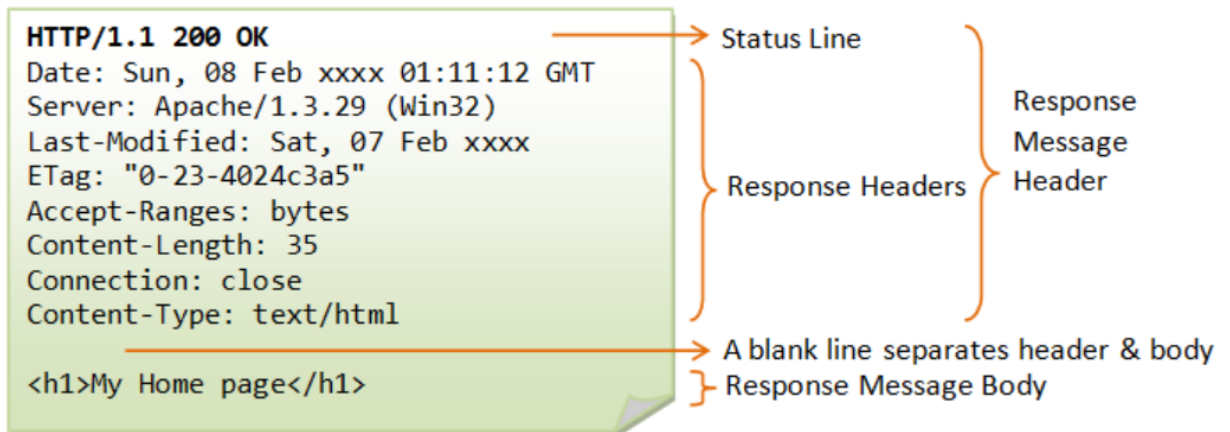
2. HTTP response

-Là thông điệp HTTP mà Server trả về cho Client thực hiện các yêu cầu của Client. Kết quả trả về là giao diện của Website và các thông tin của Header. Như vậy thông tin trả về là các đoạn mã HTML kèm theo các thông tin Header

-Cú pháp:

```
<HTTP version> <status-code> <reason-phrase>
<headers>
<blank line>
<body>
```

-Minh họa:



-Phân tích thành phần:

- **Status-line**

- HTTP version: phiên bản HTTP cao nhất mà Server hỗ trợ
- Status-code: mã kết quả trả về
- Reason-Phrase: mô tả về Status-code

- **Headers:** Hai thông tin quan trọng nhất cần cung cấp khi lập trình socket là cung cấp Content-Type và Content-Length

- **Body:** trong đề án này, body là các đoạn mã HTML

3. Cơ chế hoạt động

-Khi Client tìm kiếm một địa chỉ trên trình duyệt, trình duyệt sẽ dựa vào tên domain để gửi yêu cầu truy cập đến địa chỉ IP mà domain này đang trỏ tới.

-Lúc này request đầu tiên mà Client gửi cho server là một GET request với URL trống, không yêu cầu tài nguyên cụ thể nào

-Server nhận được request, phân tích yêu cầu, trả lời Client. Thông thường khi URL trống, Server sẽ gửi giao diện Website cho Client bằng các đoạn mã HTML

V. CHUNKED TRANSFER ENCODING

-HTTP Header "Transfer-Encoding" chỉ định hình thức mã hóa được sử dụng để truyền tải nội dung đến người dùng một cách an toàn.

-Chunked Transfer Encoding là cơ chế truyền tải nội dung có sẵn trong phiên bản 1.1 của HTTP. Trong cơ chế này, dữ liệu được chia thành các Chunk và được gửi đi một cách độc lập.

-Trong cơ chế này, độ dài dữ liệu không được xác định từ đầu. Trước hết, ta gửi Response Header trước, với "Transfer-encoding: chunked", sau đó lần lượt gửi các đoạn nội dung đi kèm với kích thước đoạn nội dung đó

-Khi gặp '0\r\n\r\n', việc gửi dữ liệu sẽ kết thúc

```
HTTP/1.1 200 OK
Content-Type: text/plain
Transfer-Encoding: chunked

7\r\n
Mozilla\r\n
9\r\n
Developer\r\n
7\r\n
Network\r\n
0\r\n
\r\n
```

B. TRIỂN KHAI ĐỒ ÁN

I. SOURCE

Mã nguồn Python

Xây dựng source code bao gồm các hàm:

- Main: tạo socket, lắng nghe client, chấp nhận kết nối, xử lý request, đóng kết nối
- Handle_request: xử lý các request của client và gửi response
- Handle_get: xử lý các GET request của client và chuẩn bị thông tin phản hồi
- Handle_post: xử lý các POST request và chuẩn bị thông tin phản hồi
- Handle_chunk: xử lý các response với dữ liệu gửi đi có kích thước lớn

II. HTML

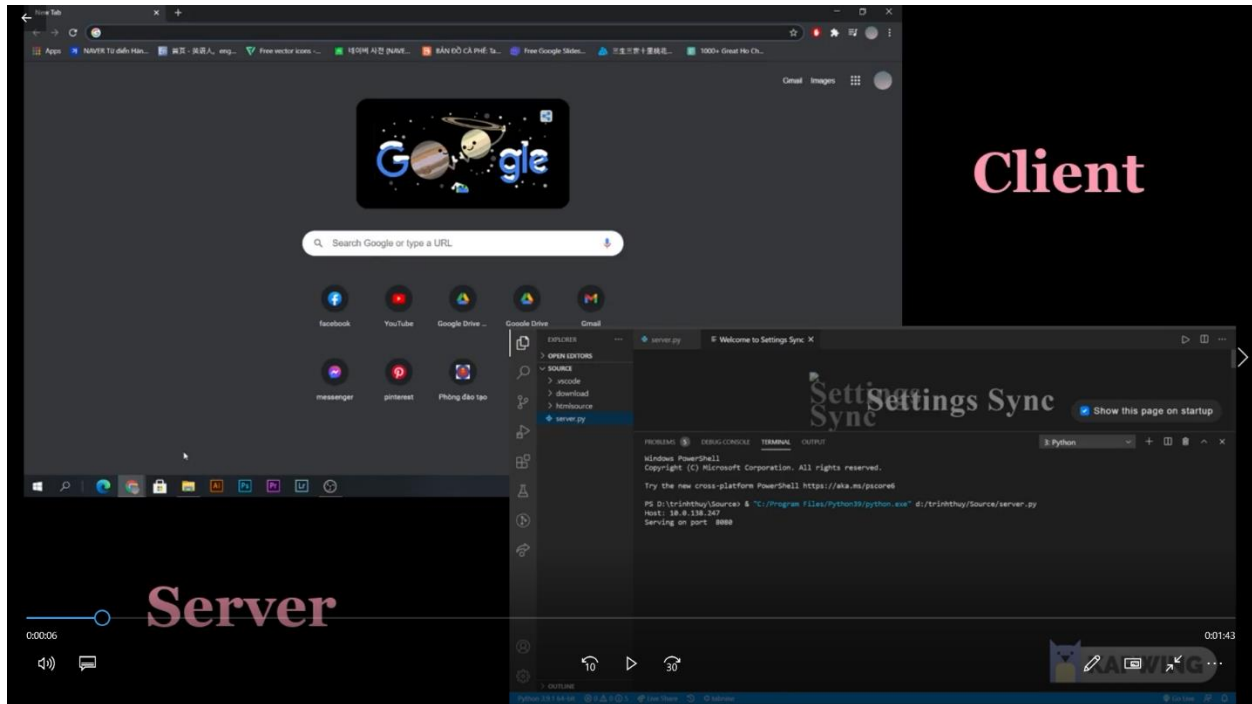
-Bao gồm các file:

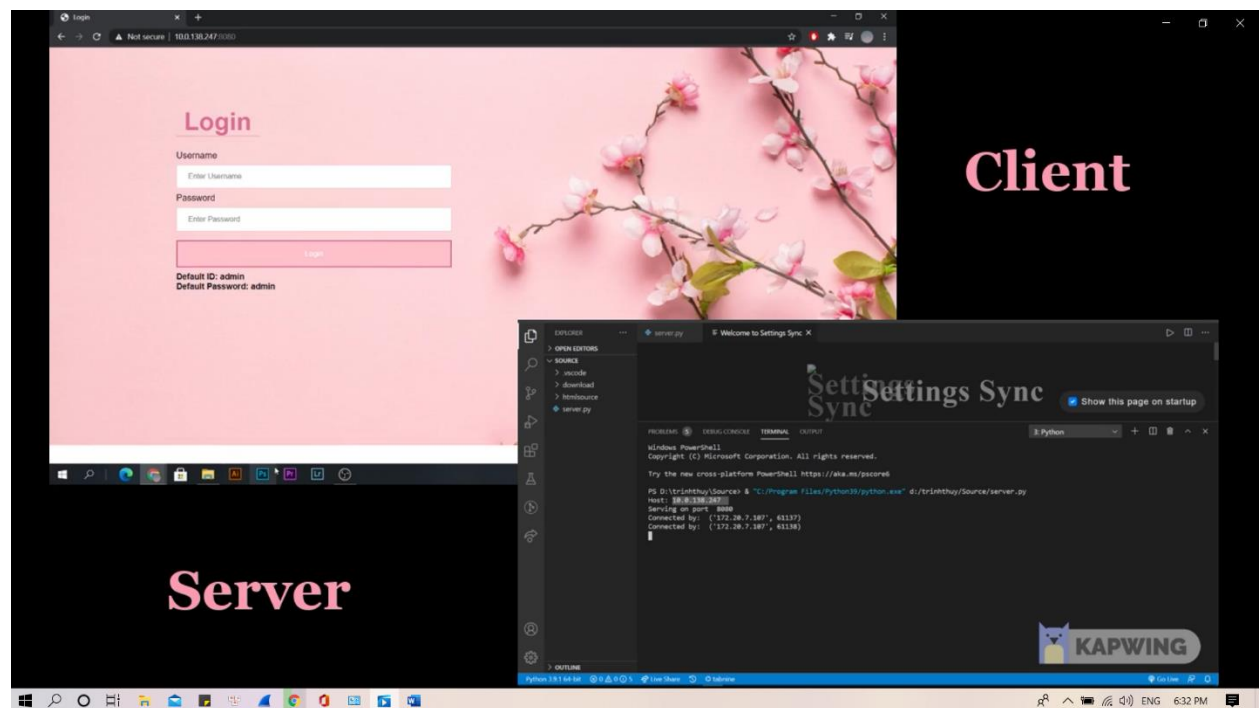
- index.html: response đầu tiên khi client truy cập, giao diện đăng nhập, username-password mặc định là admin-admin
- info.html: thông tin sinh viên thực hiện đồ án
- 404.html: khi người dùng nhập sai username và password
- file.html: các file tải về

C. DEMO

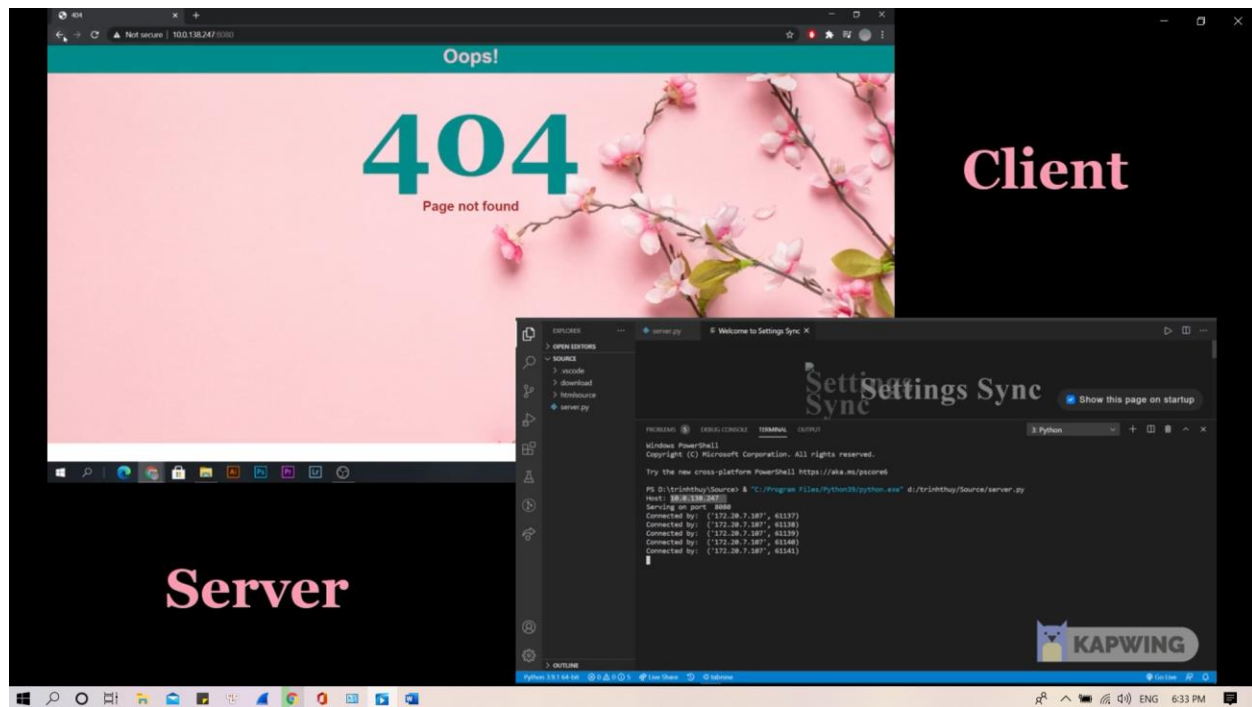
Mô tả: Laptop I đóng vai trò máy chủ (Server), Laptop II đóng vai trò máy khách(client)

Mở folder Source bằng Visual Studio Code trên Laptop I, chọn **Run python file in terminal**

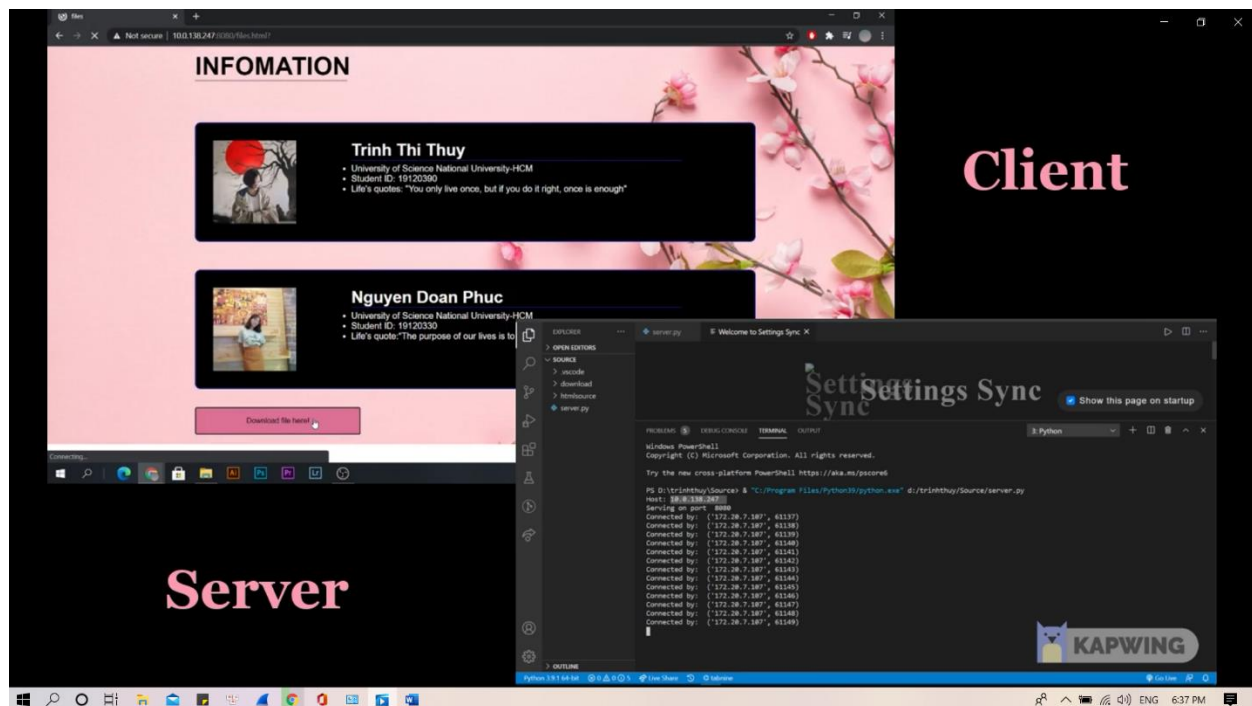




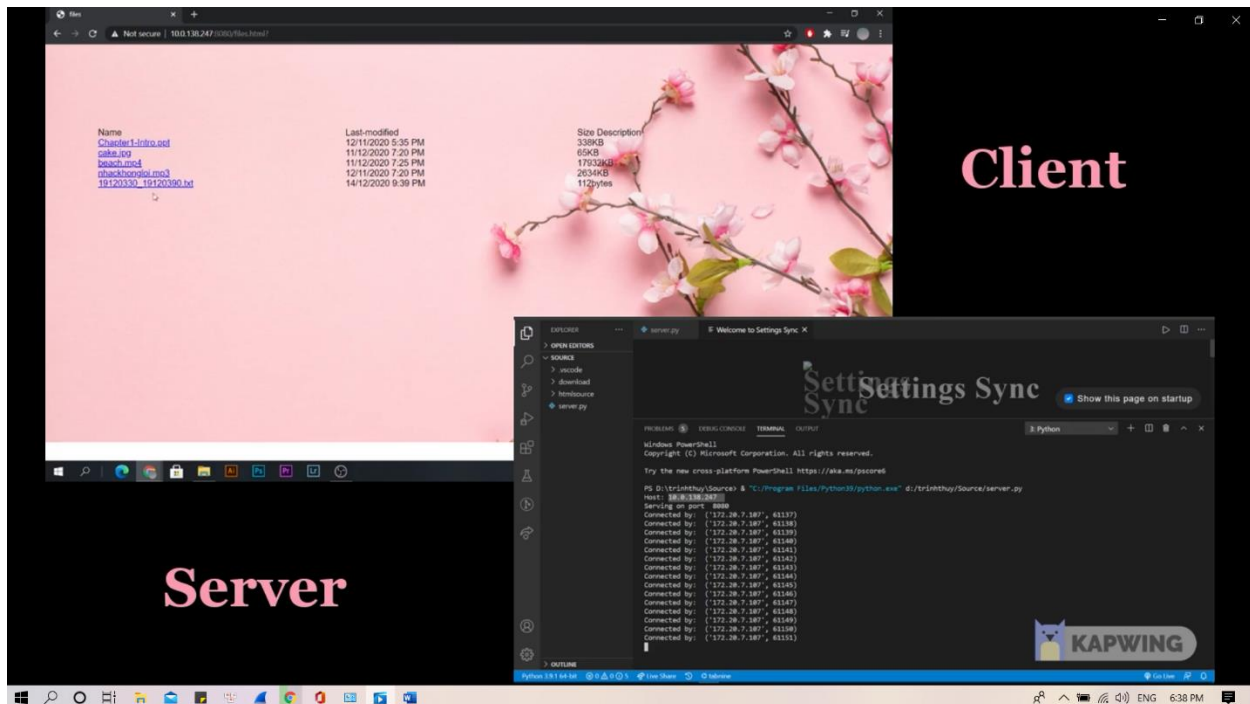
Nhập sai Password và Username thì chuyển hướng tới trang 404.html



Nhập đúng Password và Username thì chuyển hướng tới trang info.html



Bấm Download file here! để chuyển hướng tới trang files.html



Nhấp chuột vào file cần download và mở file trong thư mục của laptop II

Tài liệu, nguồn tham khảo

-<https://www.youtube.com/watch?v=OHw8OiO5v8U&feature=youtu.be>

-<https://www.youtube.com/watch?v=41XxeYkLAOk&feature=youtu.be>

-<https://drive.google.com/file/d/1A1IsrfZuzOxfEaW3ukxblOfVYUmG1T8i/view>

-<https://developer.mozilla.org/en-US/docs/Web/HTTP/Messages>

-<https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Transfer-Encoding>

