

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



ĐỒ ÁN THỰC HÀNH MẠNG MÁY TÍNH
NỘI DUNG: LẬP TRÌNH SOCKET
ĐỀ TÀI: COVID VIỆT NAM

LỚP: 20CTT2

GIẢNG VIÊN HƯỚNG DẪN:

- ĐỖ HOÀNG CƯỜNG
- LÊ HÀ MINH

THÀNH VIÊN NHÓM:

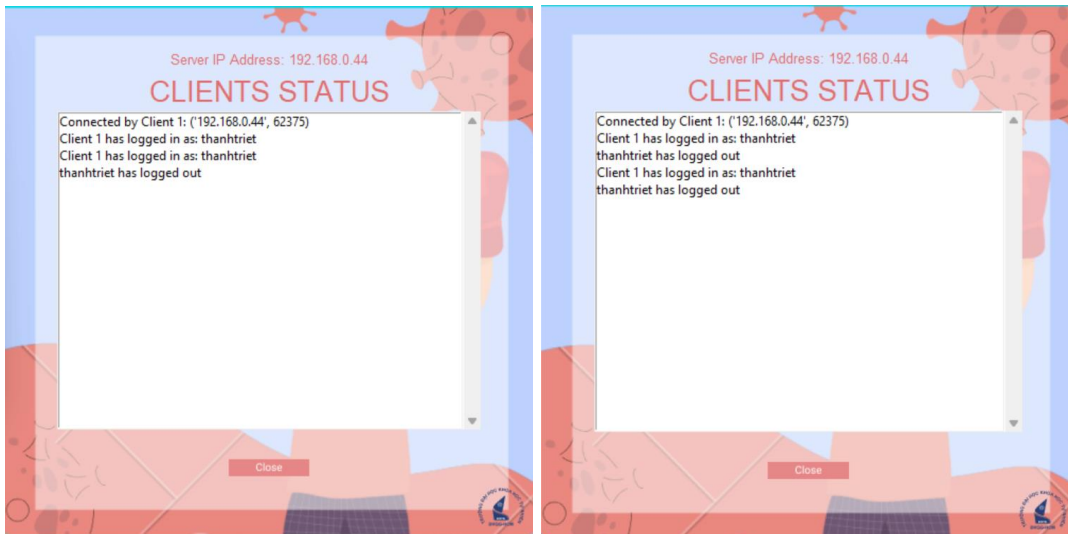
- | | |
|-------------------|----------|
| • NGUYỄN NHẬT TÂN | 20120183 |
| • LA THÀNH TRIẾT | 20120222 |
| • THÁI MINH TRIẾT | 20120223 |

MỤC LỤC

1. Đánh giá:	3
2. Kịch bản giao tiếp của chương trình:.....	3
a) Kết nối:.....	3
b) Quản lý kết nối:	3
c) Đăng nhập:	4
d) Đăng ký:	4
e) Tra cứu:	5
f) Quản lý cơ sở dữ liệu:	5
g) Thoát:	8
h) Giao diện:	9
3. Môi trường lập trình và các framework hỗ trợ:	9
4. Hướng dẫn sử dụng các tính năng chương trình:.....	10
a) Chuẩn bị:	10
b) Ứng dụng Server:	10
c) Ứng dụng Client:	11
5. Bảng phân công công việc:.....	15
a) Tổng quan:	15
b) Bảng phân công:.....	15
6. Các nguồn tài liệu tham khảo:	16

1. Đánh giá:

- **Hoàn thành:** 99%. Hoàn thành tất cả các chức năng mà đề bài yêu cầu.
- **Lỗi ở phần Giao diện:**



Một lỗi hiển thị nhỏ xảy ra ở phía Server khi Client đăng xuất rồi lại đăng nhập, bảng theo dõi trạng thái Client sẽ hiện hành động đăng nhập trước hành động đăng xuất ấy, tuy nhiên nếu thực hiện tiếp một yêu cầu khác, bảng trạng thái sẽ cập nhật lại đúng thứ tự ban đầu. Nhóm chưa tìm được cách khắc phục lỗi hiển thị này.

2. Kịch bản giao tiếp của chương trình:

Các yêu cầu của Client được gửi đến Server dưới dạng chuỗi kí tự, Server nhận và thực hiện các chức năng được yêu cầu.

a) Kết nối:

- Client nhập địa chỉ IP của Server (được ghi trên màn hình ứng dụng Server) và kết nối với Server thông qua kết nối TCP.
- Client và Server có thể đặt tại các host khác nhau, chỉ cần cùng đường mạng.
- Nếu Client nhập địa chỉ IP sai hoặc không hợp lệ, việc kết nối sẽ thất bại và nhận thông báo rằng địa chỉ IP đã nhập là không hợp lệ.
- Sau khi kết nối thành công, Client sẽ được đưa đến trang đăng nhập còn Server sẽ nhận được thông báo Client kèm địa chỉ đang kết nối.

b) Quản lý kết nối:

- Khi Client mất kết nối đột ngột, Server sẽ nhận được thông báo. Còn khi Server mất kết nối đột ngột, màn hình ứng dụng của Client vẫn sẽ được giữ nguyên, chỉ khi

Client thực hiện một yêu cầu với Server mới nhận được lỗi Server đã đóng và sau đó đóng ứng dụng.

- Mỗi khi Server nhận được kết nối từ Client, việc quản lý Client sẽ được đưa vào 1 tiểu trình riêng và khi Client này mất kết nối cũng không ảnh hưởng đến các Client khác, tạo nên một kết nối đa tiểu trình gồm nhiều Client được quản lý riêng biệt.

c) Đăng nhập:

- Tập tin Accounts.json được lưu ở Server chứa các tài khoản (gồm username và password) đã đăng ký.
- Client nhập vào username và password để gửi cho Server, phía Server nhận và tiến hành kiểm tra:
 - Nếu username đã có trong danh sách các Client đang đăng nhập, Server sẽ gửi phản hồi đã tồn tại và Client sẽ nhận được thông báo rằng tài khoản này đang được đăng nhập bởi Client khác, yêu cầu sử dụng tài khoản khác.
 - Dựa trên thông tin các tài khoản đã đăng ký được lưu bằng tập tin Accounts.json, nếu username không tồn tại, hoặc có tồn tại nhưng password không chính xác, Server sẽ gửi phản hồi sai và Client nhận được thông báo rằng thông tin nhập vào là không chính xác, yêu cầu thử lại
 - Nếu cả username và password đều trùng khớp với thông tin được lưu tại Server, Server sẽ gửi phản hồi đúng và Client được đưa tới trang tra cứu thông tin.
- Khi Client đã đăng nhập thành công, phía Server sẽ được thông báo Client nào đã đăng nhập bằng username nào và thêm username đó vào danh sách đang đăng nhập. Từ đây trạng thái của Client cũng sẽ được hiển thị bằng username.
- Khi yêu cầu đăng xuất, Client sẽ được đưa về trang đăng nhập và Server sẽ xóa username mà Client đang sử dụng khỏi danh sách đang đăng nhập.

d) Đăng ký:

- Tập tin Accounts.json được lưu ở Server chứa các tài khoản (gồm username và password) đã đăng ký.
- Client nhập username, password và nhập lại password để xác nhận, nếu trùng khớp thì tiến hành gửi cho Server, phía server nhận và tiến hành kiểm tra:
 - Dựa trên thông tin các tài khoản đã đăng ký được lưu bằng tập tin Accounts.json, nếu username đã được đăng ký, Server sẽ gửi phản hồi đã tồn tại và Client nhận được thông báo rằng username đã được đăng ký, yêu cầu sử dụng username khác.
 - Nếu username khả dụng, Server sẽ tiến hành lưu tài khoản mới vào tập tin Accounts.json, sau đó gửi phản hồi đúng và Client được đưa về trang đăng nhập để sử dụng tài khoản vừa đăng ký.

- Nếu nhập lại password không khớp với ban đầu, Client sẽ nhận được thông báo và yêu cầu nhập lại.
- Khi Client đã đăng ký thành công, phía Server sẽ được thông báo Client nào đã đăng ký bằng username nào.

e) Tra cứu:

- Tập tin TongSoCa.json (chứa thông tin Covid-19 của 63 tỉnh thành) và các tập tin dạng Ngày-Tháng-Năm.json (chứa thông tin Covid-19 của 63 tỉnh thành trong ngày hôm đó) được lấy từ một website khác và lưu về ngay khi mở Server và cứ cách 60 phút sẽ cập nhật lại các tập tin này.
- Client nhập tên tỉnh/thành phố và ngày cần tra cứu để gửi cho Server, Server nhận và tiến hành kiểm tra yêu cầu của Client:
 - Nếu Client chỉ nhập tên tỉnh/thành phố: Server tiến hành lấy tổng số ca mắc của tỉnh/thành phố đó từ tập tin TongSoCa.json để gửi cho Client. Nếu nhập sai định dạng hoặc tên tỉnh thành không tồn tại, Client sẽ nhận được thông báo lỗi và yêu cầu nhập lại.
 - Nếu Client nhập thêm ngày cần tra cứu: Server tiến hành lấy số ca mắc mới trong ngày hôm đó của tỉnh/thành phố đó từ tập tin có dạng Ngày-Tháng-Năm.json (với Ngày-Tháng-Năm là ngày mà Client đã nhập vào) để gửi cho Client. Nếu nhập sai định dạng hoặc ngày đó không tồn tại trong dữ liệu, Client sẽ nhận được thông báo lỗi và yêu cầu nhập lại.

f) Quản lý cơ sở dữ liệu:

Sử dụng file cấu trúc .json và quản lý bằng các lệnh của thư viện json, gồm các file:

Dữ liệu về thông tin tài khoản đã đăng ký (Accounts.json):

- Khi chưa có tài khoản nào:
Được tạo sẵn và có một mảng “accounts” trống để chứa các tài khoản được đăng ký

```
{
  "accounts": [
  ]
}
```

- Khi đã có tài khoản được đăng ký:
Server sẽ thêm tài khoản mới vào dưới dạng 1 list {“username”: username, “password”: password}.

```
{
  "accounts": [
    {
      "username": "thanhtriet",
      "password": "123456"
    }
  ]
}
```

Dữ liệu thông tin Covid-19 (TongSoCa.json) và Dữ liệu thông tin Covid-19 theo ngày tại các tỉnh thành (các tập tin dạng Ngày-Tháng-Năm.json):

- Kịch bản chung:

Lấy dữ liệu covid từ web bằng cách sử dụng thư viện BeautifulSoup và Selenium, sau đó ta sẽ lưu trữ dữ liệu này dưới dạng json lưu dưới server.

- Chi tiết:

1. Đầu tiên ta sẽ cài thư viện BeautifulSoup và Selenium để scrape dữ liệu từ các website theo thời gian thực
2. Sau đó ta sẽ tìm kiếm những trang web chứa dữ liệu Covid cần tìm, ở đây em sẽ dùng: [link1](#), [link2](#), [link3](#). Đây là những link web chứa dữ liệu chính thức về Covid ở Việt Nam, 3 link trên chứa toàn bộ thông tin về số ca của các tỉnh ở Việt Nam trong 1 tháng (em có thể lấy thêm dữ liệu về các tháng trước đó, nhưng ở đây em chỉ lấy dữ liệu của 1 tháng để dễ trình bày)
3. Sau đó ta sẽ bắt đầu sử dụng BeautifulSoup để scrape dữ liệu từ các link này theo ngày tháng. Ta sẽ được dữ liệu raw như sau với list tỉnh và số ca theo mỗi tỉnh:

```
13-12-2021
['Hà Nội', 'Bình Phước', 'Tây Ninh', 'Hô Chí Minh', 'Bên Tre', 'Cà Mau', 'Đồng Tháp', 'Cần Thơ', 'Khánh Hòa', 'Vĩnh Long', 'Trà Vinh', 'Sóc Trăng', 'An Giang', 'Bình Định', 'Bà Rịa - Vũng Tàu', 'Thừa Thiên Huế', 'Bạc Liêu', 'Đồng Nai', 'Kiên Giang', 'Tiền Giang', 'Hậu Giang', 'Bình Dương', 'Gia Lai', 'Đà Nẵng', 'Bắc Ninh', 'Hải Phòng', 'Thanh Hóa', 'Lâm Đồng', 'Nghệ An', 'Hưng Yên', 'Quảng Ngãi', 'Đắk Nông', 'Quảng Nam', 'Lạng Sơn', 'Hà Giang', 'Ninh Thuận', 'Long An', 'Vĩnh Phúc', 'Hà Nội', 'Thái Bình', 'Phú Yên', 'Quảng Ninh', 'Hòa Bình', 'Thái Nguyên', 'Nam Định', 'Quảng Bình', 'Hà Nam', 'Bắc Giang', 'Sơn La', 'Phú Thọ', 'Tuyên Quang', 'Lào Cai', 'Hà Tĩnh', 'Kon Tum', 'Quảng Trị', 'Yên Bái', 'Cao Bằng', 'Điện Biên', 'Lai Châu']
[1.000, '998', '919', '915', '867', '793', '740', '680', '594', '581', '476', '466', '422', '395', '371', '346', '334', '329', '328', '322', '306', '304', '268', '211', '205', '194', '183', '182', '165', '145', '134', '122', '103', '103', '96', '81', '81', '72', '58', '56', '54', '51', '49', '40', '32', '30', '30', '20', '17', '16', '15', '12', '9', '8', '8', '5', '4', '2', '1', '1']
/Users/user/Downloads/Server (4).py
```

4. Sau đó ta sẽ dùng thư viện Panda để đưa dữ liệu raw thành Dataframe. Ví dụ ngày 15-12-2021 sẽ có dữ liệu dạng Dataframe như sau:

15-12-2021		
	Tỉnh	Số Ca
0	TP. Hồ Chí Minh	1.270
1	Cà Mau	1.072
2	Bên Tre	1.035
3	Hà Nội	1.024
4	Tây Ninh	922
5	Đồng Tháp	732
6	Cần Thơ	720
7	Vĩnh Long	594
8	Khánh Hòa	593
9	Sóc Trăng	579
10	Bạc Liêu	505
11	Đồng Nai	464
12	Tiền Giang	411
13	Bình Dương	371
14	Trà Vinh	360
15	Kiên Giang	353

5. Kế đó ta sẽ lưu các Dataframe chứa dữ liệu Covid này thành file json với tên file là số ngày tháng của dữ liệu đó, ví dụ ngày 15-12-2021 sẽ có dữ liệu Covid dưới dạng json như sau:

```
{ } 15-12-2021.json > ...  
1  {  
2    "TP. H\ued3 Ch\ued Minh": "1.270",  
3    "C\ued Mau": "1.072",  
4    "B\uebfm Tre": "1.035",  
5    "H\ued N\ued9i": "1.024",  
6    "T\ued2y Ninh": "922",  
7    "\u0110\ued3ng Th\ued1p": "732",  
8    "C\uea7n Th\ued1a1": "720",  
9    "V\ued129nh Long": "594",  
10   "Kh\ued1nh H\uedf2a": "593",  
11   "S\uedf3c Tr\ued103ng": "579",  
12   "B\uea1c Li\uedeau": "505",  
13   "\u0110\ued3ng Nai": "464",  
14   "Ti\uec1n Giang": "411",  
15   "B\uedecnh D\ued1b0\ued1a1ng": "371",  
16   "Tr\ued0e0 Vinh": "360",
```

6. Chương trình sẽ tiếp tục scrape tất cả các link và ta sẽ có một list các file json của những ngày khác nhau (như hình)

```
{ } 01-12-2021.json  
{ } 02-12-2021.json  
{ } 03-12-2021.json  
{ } 04-12-2021.json  
{ } 05-12-2021.json  
{ } 06-12-2021.json  
{ } 07-12-2021.json  
{ } 08-12-2021.json  
{ } 09-12-2021.json
```

7. Đó là dữ liệu về số ca mắc mới của từng tỉnh theo ngày, tuy nhiên chúng ta lại thiếu dữ liệu Tổng số ca mắc của từng tỉnh nhưng mà những trang web trên lại không có nên em sẽ scrape thêm trang web khác để lấy dữ liệu. Trang web em sẽ dùng: [Tổng Số Ca mắc của từng tỉnh](#)

8. Do dữ liệu trong web trên được load bằng JavaScript hay nói cách khác đây là Dynamic Website nên em không thể dùng thư viện BeautifulSoup như ở trên nên em sẽ dùng thư viện khác tên là Selenium. Thư viện này sẽ giúp em scrape được dữ liệu từ các Dynamic Websites
9. Chúng ta sẽ cài đặt thư viện Selenium và tiến hành tương tự như trên, sau đó ta sẽ xuất dữ liệu raw thành Dataframe và lưu thành file json

```
{ } TongSoCa.json > ...
1  [
2    "C\u1ea3n \u01b0\u1edbc[a]": "1.347.201",
3    "H\u1ed3 Ch\u00ed Minh": "502.632",
4    "B\u00ecnh D\u01b0\u01a1ng": "290.513",
5    "\u0110\u1ed3ng Nai": "97.418",
6    "T\u00e2y Ninh": "74.058",
7    "\u0110\u1ed3ng Th\u00e0i": "43.077",
8    "H\u00e0 N\u1ed9i": "42.412",
9    "C\u1ea7n Th\u01a1": "40.322",
10   "Long An": "40.297",
11   "Ti\u1ec1n Giang": "33.282",
12   "C\u00e0 Mau": "32.718",
13   "An Giang": "32.011",
```

File TongSoCa.json này chứa dữ liệu về tổng số ca mắc của từng tỉnh

- Việc lấy thông tin và tạo các tập tin này sẽ được tách ra thành một tiểu trình riêng, cứ mỗi 60 phút sẽ cập nhật lại một lần.

g) Thoát:

- Client gửi thông báo cho Server, sau đó thoát chương trình, Server sẽ nhận được thông báo Client nào đã ngắt kết nối. Thông báo sẽ khác nhau khi Client sử dụng chức năng thoát của chương trình và khi Client đột ngột mất kết nối.
- Khi đóng Server, Client chưa nhận được thông báo ngay, tới khi Client thực hiện yêu cầu nào đó thì mới được thông báo rằng Server đã đóng để tránh trường hợp Client cần ghi chép lại thông tin đang hiển thị trên màn hình, tương tự như khi Server mất kết nối đột ngột. Thông báo này được gửi dưới dạng báo lỗi.

h) Giao diện:

- Sử dụng thư viện tkinter để thiết kế giao diện.
- Giao diện lựa chọn màu sắc và hình ảnh sáng, bắt mắt thể hiện một tinh thần lạc quan, tự tin chiến thắng đại dịch.

Giao diện của ứng dụng Client gồm 4 trang:

- **Trang Kết nối:**
 - Trang đầu tiên khi mở ứng dụng, gồm 1 ô nhập địa chỉ IP của Server và 1 nút có chức năng kết nối tới địa chỉ IP này.
 - Nếu kết nối thành công, ứng dụng sẽ chuyển sang trang đăng nhập, nếu không sẽ hiện dòng thông báo địa chỉ IP không hợp lệ.
- **Trang Đăng nhập:**
 - Gồm 2 ô nhập username và password, 1 nút có chức năng đăng nhập, 1 nút có chức năng chuyển ứng dụng sang trang đăng ký và 1 nút có chức năng thoát.
 - Nếu đăng nhập thành công, ứng dụng sẽ chuyển sang trang tra cứu, nếu không sẽ hiện dòng thông báo lỗi mà người dùng đang gặp.
- **Trang Đăng ký:**
 - Gồm 3 ô nhập username, password và nhập lại password, 1 nút có chức năng đăng ký và 1 nút có chức năng quay trở lại trang đăng nhập.
 - Nếu đăng ký thành công, ứng dụng sẽ chuyển về trang đăng nhập và hiển thị thông báo đăng ký thành công, nếu không sẽ hiện dòng thông báo lỗi mà người dùng đang gặp.
- **Trang Tra cứu:**
 - Gồm 2 ô nhập tên tỉnh thành và ngày tháng, 1 nút có chức năng tra cứu và 1 nút có chức năng đăng xuất.
 - Nếu tra cứu thành công, ứng dụng sẽ hiển thị thông tin được yêu cầu, nếu không sẽ hiện dòng thông báo lỗi mà người dùng đang gặp.

Giao diện của ứng dụng Server chỉ gồm một trang duy nhất, gồm:

- 1 dòng hiển thị địa chỉ IP của Server
- 1 bảng hiển thị các yêu cầu mà Client đã thực hiện thành công
- 1 nút có chức năng đóng Server

3. Môi trường lập trình và các framework hỗ trợ:

- **Môi trường lập trình:** Python 3.7
- **IDE:** Visual Studio 2019
- **Các thư viện hỗ trợ lấy dữ liệu từ website:** bs4, requests, selenium, lxml.

4. Hướng dẫn sử dụng các tính năng chương trình:

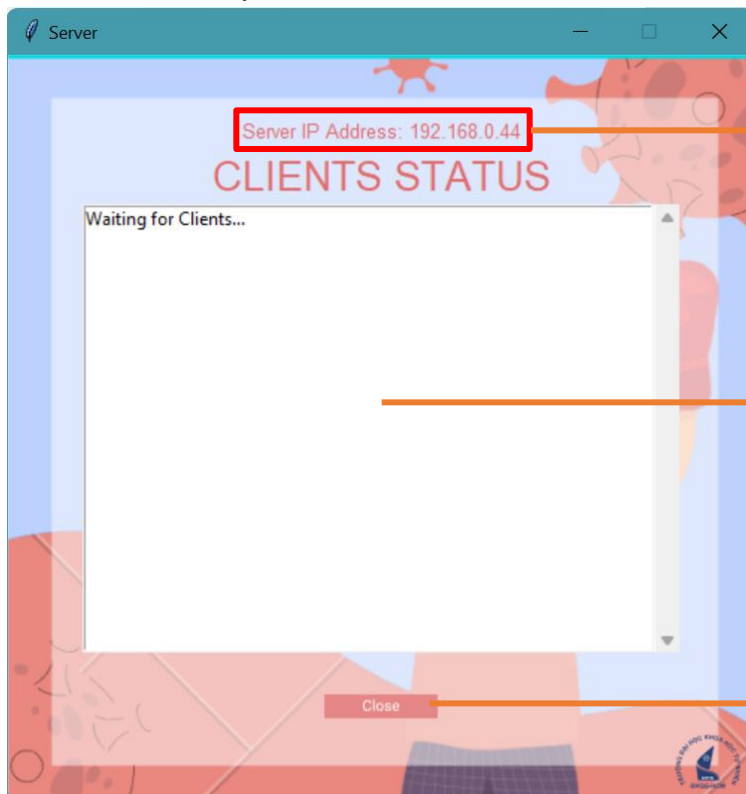
a) Chuẩn bị:

Để có thể chạy chương trình, cần:

- Cài đặt môi trường Python 3.7 hoặc mới hơn
- Cài đặt các thư viện cần thiết: bs4, requests, selenium, lxml (có thể cài bằng pip)

b) Ứng dụng Server:

- Tại thư mục Source, vào thư mục Server (chứa file Server.py, Accounts.json và các hình ảnh của giao diện).
- Mở file Server.py
- Màn hình làm việc chính của Server:



Địa chỉ IP của Server mà Client sử dụng để kết nối.

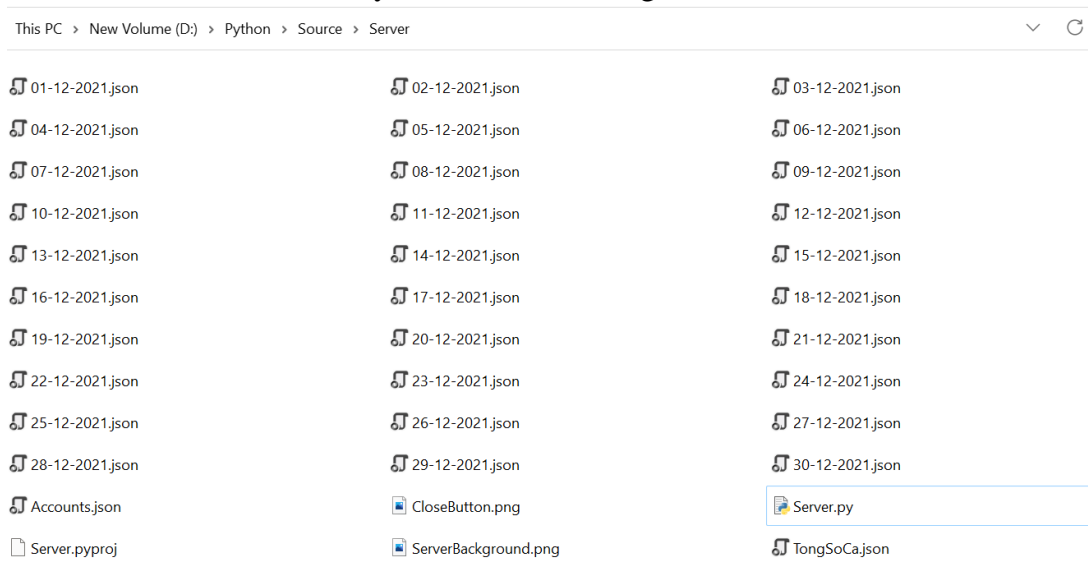
Nơi hiển thị các hoạt động của các Client

Nút đóng Server

- Khi đã có Client kết nối:

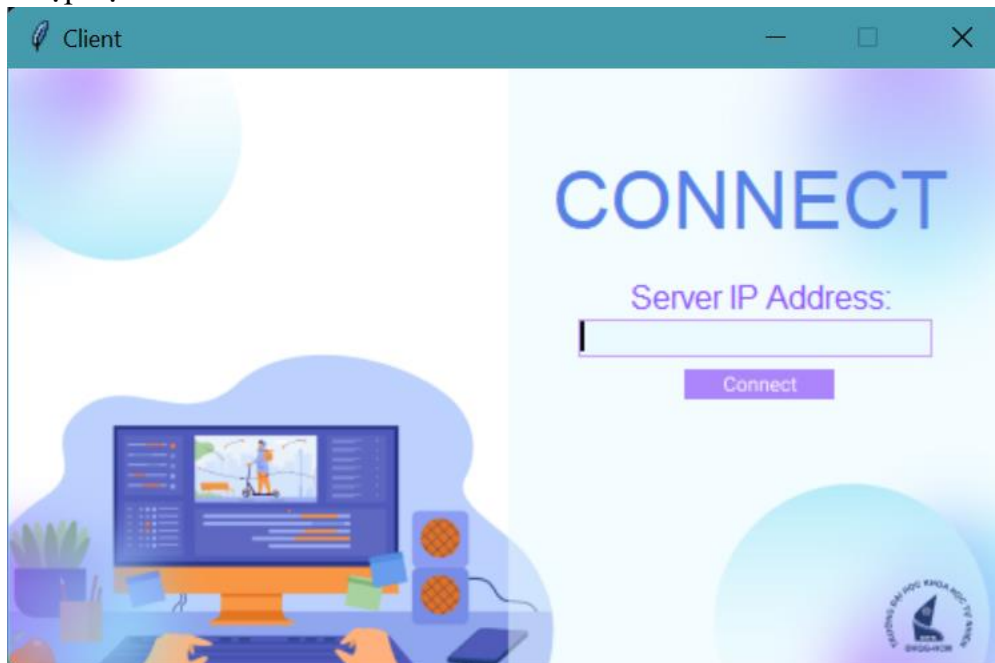


- Thư mục Server sau khi lấy dữ liệu thành công:



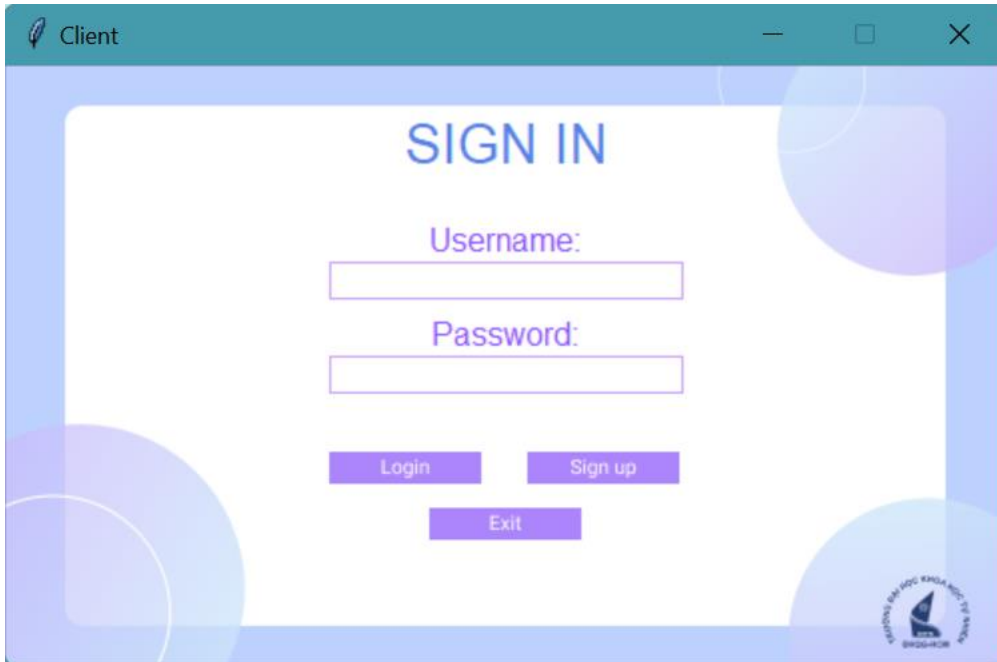
c) Ứng dụng Client:

- Tại thư mục Source, vào thư mục Client (chứa file Client.py và các hình ảnh của giao diện).
- Mở file Client.py, trang kết nối sẽ xuất hiện.
- Nhập địa chỉ IP của Server và nhấn vào nút Connect



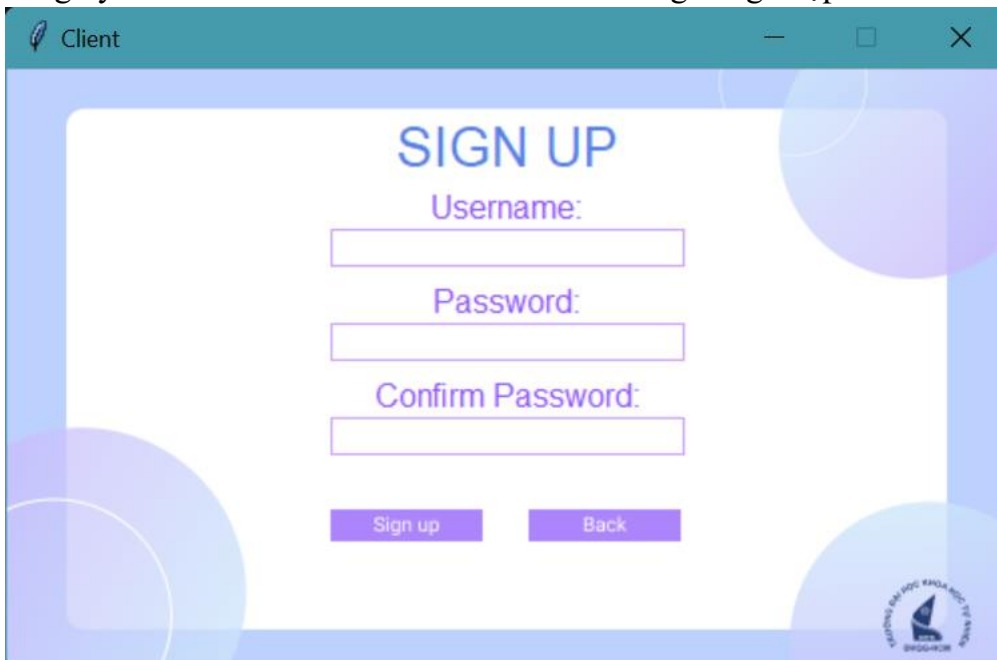
- Sau khi kết nối thành công sẽ chuyển sang trang đăng nhập, nếu đã có tài khoản hãy nhập vào ô Username và Password sau đó nhấn nút Login để đăng nhập vào trang

Tra cứu, nếu chưa có tài khoản thì nhấn vào nút Sign up để chuyển tới trang đăng ký tài khoản.



The image shows a software window titled "Client" with a teal header bar. The main content area has a light blue background with abstract circular patterns. At the top center, the text "SIGN IN" is displayed in a large, bold, blue font. Below this, there are two input fields: "Username:" and "Password:", each with a corresponding text box. Under the input fields, there are three buttons: "Login" and "Sign up" are side-by-side, and "Exit" is centered below them. All buttons are blue with white text. In the bottom right corner, there is a small circular logo with a sailboat and the text "TRƯỜNG ĐẠI HỌC KHOA HỌC VÀ CÔNG NGHỆ" and "SHOOS-HUB" below it.

- Tại trang đăng ký, hãy điền vào ô Username và Password, xác nhận lại password tại ô Confirm Password, sau đó nhấn vào nút Sign up để đăng ký. Nếu đăng ký thành công chương trình sẽ chuyển về trang đăng nhập để đăng nhập. Nếu không muốn đăng ký nữa có thể nhấn nút Back để trở về trang đăng nhập.



The image shows a software window titled "Client" with a teal header bar. The main content area has a light blue background with abstract circular patterns. At the top center, the text "SIGN UP" is displayed in a large, bold, blue font. Below this, there are three input fields: "Username:", "Password:", and "Confirm Password:", each with a corresponding text box. Under the input fields, there are two buttons: "Sign up" and "Back", side-by-side. All buttons are blue with white text. In the bottom right corner, there is a small circular logo with a sailboat and the text "TRƯỜNG ĐẠI HỌC KHOA HỌC VÀ CÔNG NGHỆ" and "SHOOS-HUB" below it.

- Tại trang tra cứu:
- Nếu muốn tra cứu tổng số ca mắc của một tỉnh/thành phố, hãy nhập tên tỉnh/thành phố đó vào ô Province dưới dạng viết hoa chữ cái đầu có dấu, VD: An Giang (riêng thành phố Hồ Chí Minh thì nhập TP. Hồ Chí Minh) và bỏ trống ô Date, sau đó nhấn vào nút Look up



The screenshot shows a web application window titled "Client". The main heading is "COVID-19". Below it, there is a "Province" label and a text input field containing "TP. Hồ Chí Minh". Underneath is a "Date" label and an empty text input field. There are two buttons: "Look up" and "Log out". At the bottom, it displays "Total cases: 504.024". The interface is decorated with purple virus-like icons in the corners.

- Nếu muốn tra cứu số ca mắc trong một ngày cụ thể của tỉnh/thành phố đó, hãy nhập thêm vào ô Date dưới dạng dd-mm-yy, VD: 01-12-2021, sau đó nhấn vào nút Look up



The screenshot shows the same web application window. The "Province" field still contains "TP. Hồ Chí Minh". The "Date" field now contains "01-12-2021". The "Look up" and "Log out" buttons are still present. At the bottom, it displays "New cases (01-12-2021): 1.675". The interface is decorated with purple virus-like icons in the corners.

- Nếu muốn đăng xuất, nhấn nút Log out để đăng xuất và trở về trang đăng nhập.
- Tại trang đăng nhập nếu muốn ngắt kết nối với Server và thoát chương trình, nhấn nút Exit.
- Tất cả các trường hợp yêu cầu của người dùng thất bại đều sẽ nhận được thông báo và lý do thực hiện thất bại tại các trang của ứng dụng



- Nếu Server đóng hoặc mất kết nối, một cửa sổ lỗi sẽ hiện lên để thông báo sau khi người dùng thực hiện một yêu cầu đối với Server và đóng ứng dụng



5. Bảng phân công công việc:

a) Tổng quan:

- Nguyễn Nhật Tân (20120183): Tra cứu thông tin sử dụng dữ liệu từ website khác, cho phép tra cứu theo ngày. Dựa trên thang điểm: 3.5 điểm
- 6.5 điểm còn lại được chia thành 2 phần:
 - Xây dựng kịch bản, ý tưởng, các bước giao tiếp giữa Client và Server, các cơ sở dữ liệu, đặt ra các trường hợp kiểm thử chương trình, hỗ trợ tìm lỗi, thiết kế giao diện: Thái Minh Triết (20120223)
 - Tìm hiểu cách thức và lập trình: La Thành Triết (20120222)

b) Bảng phân công:

	Chức năng	Ý nghĩa	Mở rộng
1	Kết nối	Kết nối cơ bản: La Thành Triết	Client có thể nhập IP và kết nối với Server đặt tại host khác: - Ý tưởng: Thái Minh Triết - Lập trình: La Thành Triết
2	Quản lý kết nối	Khi mất kết nối đột ngột không làm chương trình treo hay lỗi: - Kịch bản: Thái Minh Triết - Lập trình: La Thành Triết	Quản lý kết nối đa tiểu trình: - Kịch bản: Thái Minh Triết - Lập trình: La Thành Triết
3	Đăng nhập	- Kịch bản và các bước giao tiếp: Thái Minh Triết - Lập trình: La Thành Triết	
4	Đăng ký	- Kịch bản và các bước giao tiếp: Thái Minh Triết - Lập trình: La Thành Triết	
5	Tra cứu	- Cho phép tra cứu theo ngày: Nguyễn Nhật Tân - Hỗ trợ ghép source code vào chương trình chính: La Thành Triết	Lấy dữ liệu từ website khác để tra cứu: Nguyễn Nhật Tân Cập nhật dữ liệu: - Ý tưởng: Thái Minh Triết - Lập trình: La Thành Triết
6	Quản lý cơ sở dữ liệu	- Dữ liệu phân tra cứu: Nguyễn Nhật Tân - Dữ liệu phân đăng nhập, đăng ký: La Thành Triết	
7	Thoát	- Ý tưởng: Thái Minh Triết - Lập trình: La Thành Triết	
8	Giao diện		Có thiết kế giao diện đồ họa cho chương trình (GUI):

			- Hình ảnh và ý tưởng: Thái Minh Triết - Lập trình: La Thành Triết
--	--	--	---

6. Các nguồn tài liệu tham khảo:

- Tài liệu môn học được đăng ở Moodle
- Video hướng dẫn của giảng viên
- [\[Mạng máy tính - HCMUS\] Đồ án lập trình Socket - Python | Local server - YouTube](#)
- [Python Socket Programming Tutorial - YouTube](#)
- [Cùng Python Crawl Dữ Liệu Để Tạo Bài Báo Nhanh \(codelearn.io\)](#)