

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



# BÁO CÁO BỘ MÔN MẠNG MÁY TÍNH

**Giáo viên hướng dẫn**

**ThS. Lê Hà Minh**

**Đề tài: Lập trình web server**

Thành Phố Hồ Chí Minh – 2020



# MỤC LỤC:

1. Phân công công việc.
2. Mô tả thuật toán và cách hoạt động.
3. Demo các bước thực hiện.



# **I. Phân Công Công Việc:**

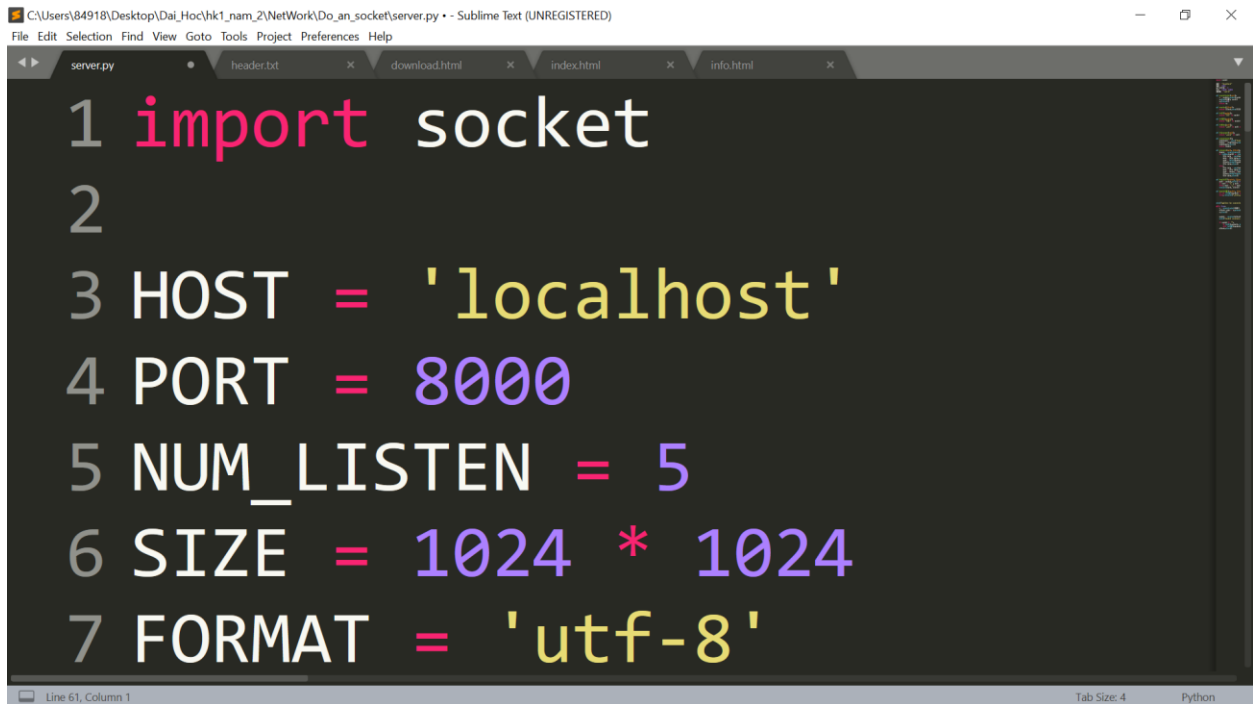
- Trần Đông Ba (MSSV: 19127334):
  - + Tìm hiểu thuật toán và code.
  - + Tìm hiểu về python, html.
  - + Xây dựng code mẫu.
  
- Kiều Hải Đăng (MSSV: 19127347):
  - + Tìm hiểu thuật toán và code.
  - + Tối ưu code.
  - + Viết báo cáo mô tả các bước thực hiện thuật toán, demo web server.



- Tiến độ công việc:
  - + Hoàn thành ngày 4/12/2020.
  - + Mọi thành viên tích cực tìm hiểu và xây dựng web server đúng thời hạn.



## II. Mô tả thuật toán và cách hoạt động:



```
1 import socket
2
3 HOST = 'localhost'
4 PORT = 8000
5 NUM_LISTEN = 5
6 SIZE = 1024 * 1024
7 FORMAT = 'utf-8'
```

The screenshot shows a Sublime Text editor window with the title bar "C:\Users\84918\Desktop\Đai\_Hoc\hkt1\_nam\_2\NetWork\Do\_an\_socket\server.py - Sublime Text (UNREGISTERED)". The editor has several tabs open: "server.py", "header.txt", "download.html", "index.html", and "info.html". The "server.py" tab is active, displaying the Python code shown in the code block. The status bar at the bottom indicates "Line 61, Column 1", "Tab Size: 4", and "Python".

Trước hết ta tiến hành khai báo một số hằng số cơ bản cần thiết cho việc tạo socket.



```
C:\Users\84918\Desktop\Đai_Hoc\hk1_nam_2\NetWork\Do_an_socket\server.py - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

server.py  header.txt  download.html  index.html  info.html
64 print('waiting for connection...')
65
66 while True:
67     s = createSocket(PORT)
68     client, addr = s.accept()
69     s.close()
70
71     packet = receive(client)
72     print('packet received : ', packet)
73
74     if packet != '':
75         if isGET(packet): handleGET(packet, client)
76         elif isPOST(packet): handlePOST(packet, client)
77     client.close()
```

Đầu tiên ta sẽ tạo ra một socket bằng hàm createSocket do ta viết, truyền vào cổng ta muốn mở để server sử dụng cho việc truyền/nhận các gói tin.



```
C:\Users\84918\Desktop\Dat_Hoc\hk1_nam_2\NetWork\Do_an_socket\server.py - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
server.py header.txt download.html index.html info.html
3 HOST = 'localhost'
4 PORT = 8000
5 NUM_LISTEN = 5
6 SIZE = 1024 * 1024
7 FORMAT = 'utf-8'
8
9 def createSocket(port):
10     s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
11     s.bind((HOST, port))
12     s.listen(5)
13     return s;
14
15 def receive(Client):
16     return Client.recv(SIZE).decode(FORMAT)
17
```

Ta dùng hàm socket được tích hợp sẵn trong thư viện socket để tạo ra một socket với :

+ AF\_INET là IPV4.

+ SOCK\_STREAM là TCP.

Tiếp đó ta tiến hành bind để xác nhận socket và mở port trên máy tính. Với lưu lượng truy cập tối đa là 5 client. Sau đó ta trả về một socket để sử dụng.



```
C:\Users\84918\Desktop\Đai_Hoc\hk1_nam_2\NetWork\Do_an_socket\server.py - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

server.py  header.txt  download.html  index.html  info.html

64 print('waiting for connection...')
65
66 while True:
67     s = createSocket(PORT)
68     client, addr = s.accept()
69     s.close()
70
71     packet = receive(client)
72     print('packet received : ', packet)
73
74     if packet != '':
75         if isGET(packet): handleGET(packet, client)
76         elif isPOST(packet): handlePOST(packet, client)
77     client.close()
```

Line 13, Column 14      Tab Size: 4      Python

Ta khai báo hai biến là client tức là client hiện tại ta sử dụng và biến addr lưu trữ địa chỉ của socket. Sau đó ta đóng socket lại.





```
C:\Users\84918\Desktop\Đai_Hoc\hk1_nam_2\NetWork\Do_an_socket\server.py - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

server.py  header.txt  download.html  index.html  info.html

64 print('waiting for connection...')
65
66 while True:
67     s = createSocket(PORT)
68     client, addr = s.accept()
69     s.close()
70
71     packet = receive(client)
72     print('packet received : ', packet)
73
74     if packet != '':
75         if isGET(packet): handleGET(packet, client)
76         elif isPOST(packet): handlePOST(packet, client)
77     client.close()
```

Sau khi đã nắm được client (tức là data của nó) ta lưu nó vào biến tên là packet. Hàm receive được implement như bên dưới.



```
C:\Users\84918\Desktop\Đai_Hoc\hk1_nam_2\NetWork\Do_an_socket\server.py - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

server.py header.txt download.html index.html info.html
10 s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
11 s.bind((HOST, port))
12 s.listen(5)
13 return s;
14
15 def receive(Client):
16     return Client.recv(SIZE).decode(FORMAT)
17
18 def isGET(packet):
19     return 'GET' in packet
20
```

Ta trả về data do client nắm giữ với SIZE ta đã khai báo ở trên và decode theo format tiêu chuẩn là utf-8. Rồi ta in thông tin ta nhận được ra console để kiểm tra.



```
C:\Users\84918\Desktop\Doan\hoc\k1_nam_2\NetWork\Do_an_socket\server.py - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

server.py  header.txt  download.html  index.html  info.html

64 print('waiting for connection...')
65
66 while True:
67     s = createSocket(PORT)
68     client, addr = s.accept()
69     s.close()
70
71     packet = receive(client)
72     print('packet received : ', packet)
73
74     if packet != '':
75         if isGET(packet): handleGET(packet, client)
76         elif isPOST(packet): handlePOST(packet, client)
77     client.close()

Line 13, Column 14 Tab Size: 4 Python
```

```
C:\Users\84918\Desktop\New folder\Đồ Án - Copy\Server.py - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

Server.py  header.txt  download.html  index.html  info.html

33
34 def isGetRequest(packet):
35     return 'GET' in packet
36
37
38 def isPostRequest(packet):
39     return 'POST' in packet
40
41

Line 1, Column 1 Spaces: 4 Python
```

Thành Phố Hồ Chí Minh – 2020



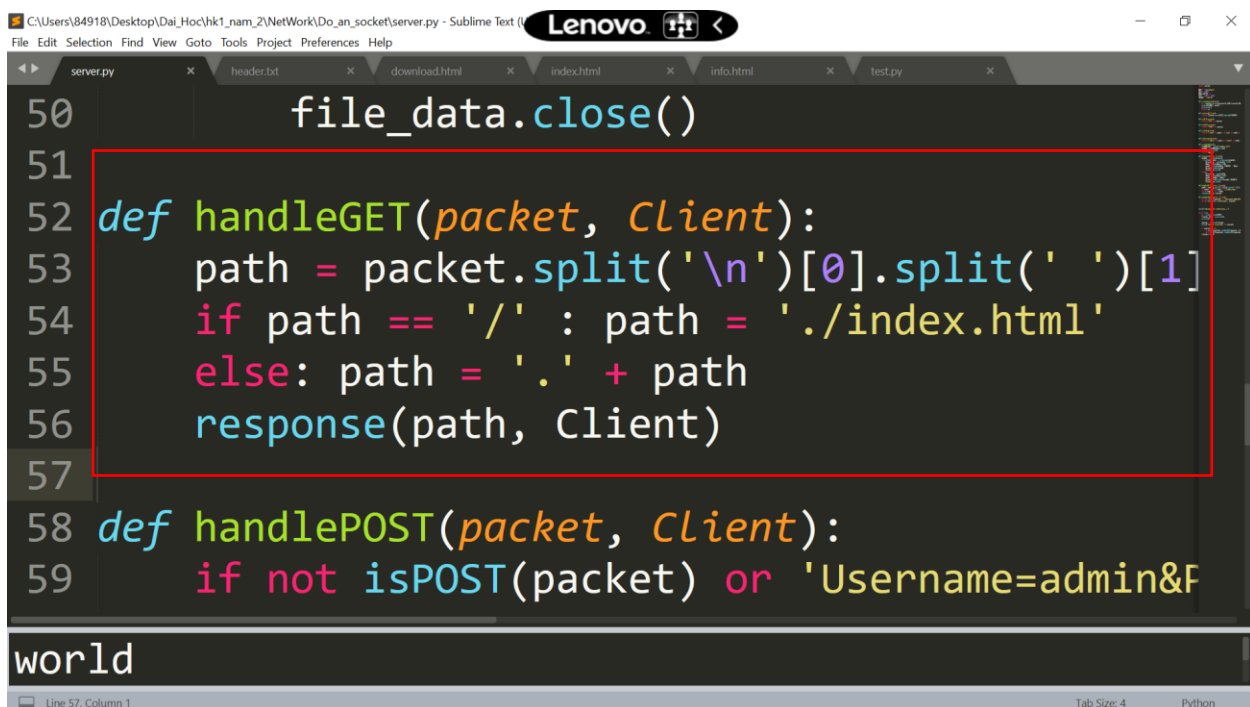
Để chắc chắn ta nên kiểm tra xem packet(tức là data) nhận được từ client đưa lên là không rỗng.

Với hai method chính là:

+ POST

+ GET

Ta tiến hành kiểm tra xem nếu trình duyệt yêu cầu GET ta sẽ xử lý GET ngược lại ta xử lý POST. Để biết là GET hay POST ta tự viết hàm để kiểm tra sự xuất hiện của GET hoặc POST trong packet.

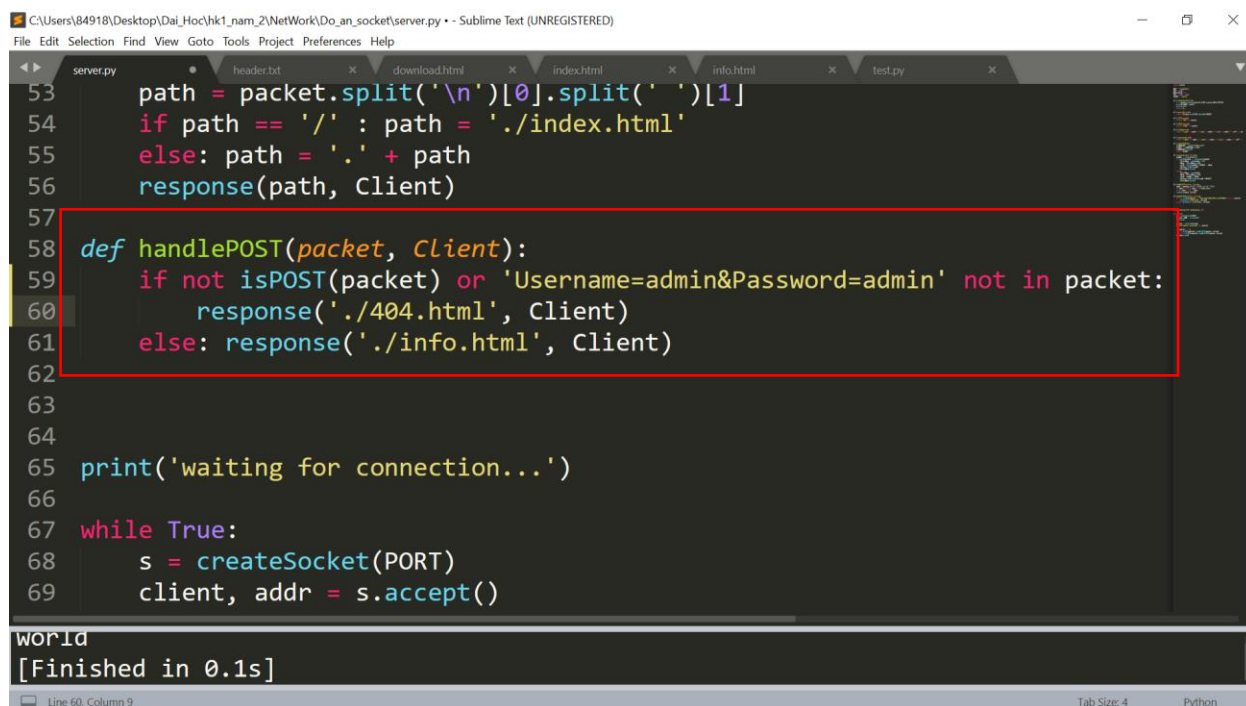


```
50 file_data.close()
51
52 def handleGET(packet, Client):
53     path = packet.split('\n')[0].split(' ')[1]
54     if path == '/' : path = './index.html'
55     else: path = '.' + path
56     response(path, Client)
57
58 def handlePOST(packet, Client):
59     if not isPOST(packet) or 'Username=admin&F
```

world



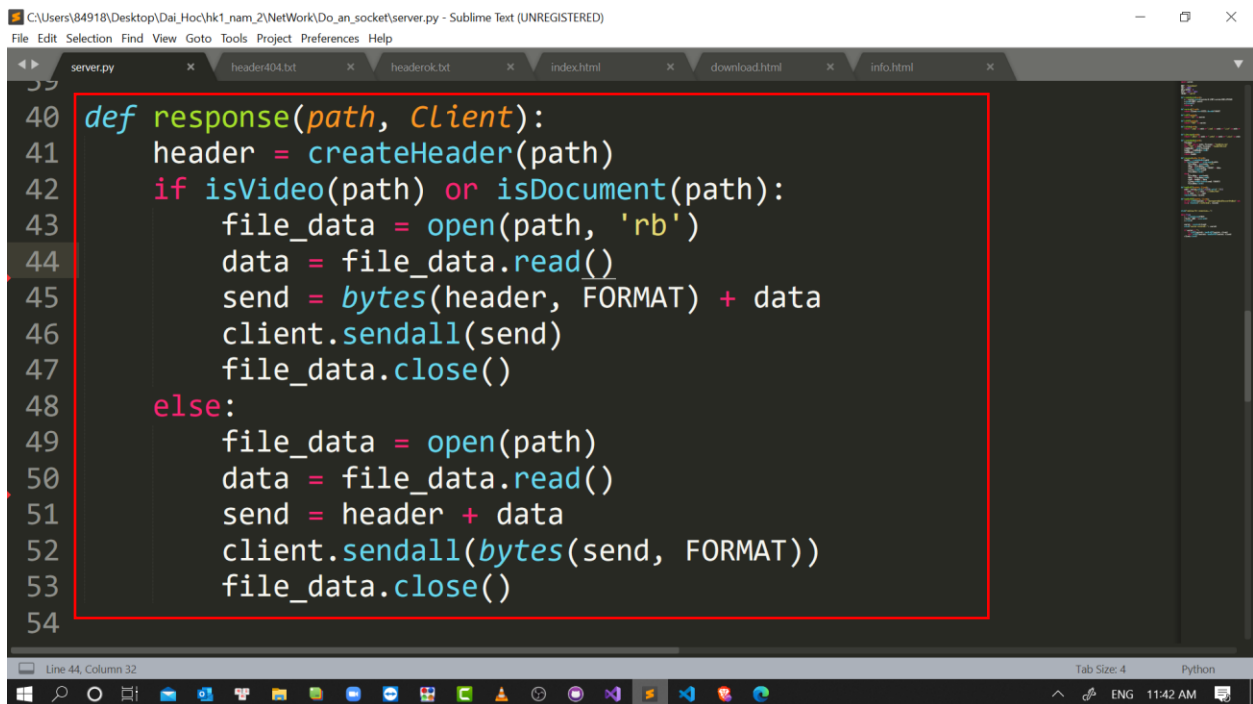
`path = packet.split('\n')[0].split(' ')[1]` ở dòng code này tức là khi ta có data là 1 dòng header với dòng đầu là `HTTP / 1.1 200 OK` v.v..thì đầu tiên ta chia chuỗi này thành các chuỗi nhỏ sau mỗi dấu `\n` thì khi cắt xong ta được chuỗi như sau : `HTTP / 1.1 200 OK` tiếp đó từ chuỗi này ta tiếp tục cắt ra thành các chuỗi con ngăn cách nhau bởi dấu `' '` (khoảng trắng). ta được từng phần tử như sau `{HTTP, /, 1.1, 200, OK}` sau đó lấy phần tử tại vị trí số 1 tức là dấu `'/'` vậy là ta được phần đường dẫn tới thư mục. ta lưu vào biến `path` để tiện sử dụng. Sau đó tiến hành kiểm tra nếu `path` là `'/'` tức là khi vừa với vào và kết nối tới server thì ta sẽ thay đường dẫn để client đi đến trang `index.html` còn ngược lại nếu phần ta split được không phải `'/'` mà là `./info.html` thì ta cộng vào đó dấu `'.'` để biến thành đường dẫn không cố định đến thư mục (vì không phải máy nào cũng có đường dẫn như nhau). Rồi ta gọi hàm `response` để nó trả về nội dung của file ta vừa mở cho client.



```
server.py
53 path = packet.split('\n')[0].split(' ')[1]
54 if path == '/' : path = './index.html'
55 else: path = '.' + path
56 response(path, Client)
57
58 def handlePOST(packet, Client):
59     if not isPOST(packet) or 'Username=admin&Password=admin' not in packet:
60         response('./404.html', Client)
61     else: response('./info.html', Client)
62
63
64
65 print('waiting for connection...')
66
67 while True:
68     s = createSocket(PORT)
69     client, addr = s.accept()
70
71 world
72 [Finished in 0.1s]
```



Tương tự với GET thì ta cũng có hàm xử lý nếu method là POST. Đầu tiên ta tiến hành kiểm tra nếu nó không có POST xuất hiện trong packet hoặc Username hay Password không phải admin (có thể không nhập nếu không để thẻ required) thì ta gọi hàm response để trả về nội dung trang 404.html, ngược lại thì ta trả về nội dung trang info.html cho client.



```
40 def response(path, Client):
41     header = createHeader(path)
42     if isVideo(path) or isDocument(path):
43         file_data = open(path, 'rb')
44         data = file_data.read()
45         send = bytes(header, FORMAT) + data
46         client.sendall(send)
47         file_data.close()
48     else:
49         file_data = open(path)
50         data = file_data.read()
51         send = header + data
52         client.sendall(bytes(send, FORMAT))
53         file_data.close()
54
```

Hàm response được xây dựng như sau:

Đầu tiên ta tạo ra header bằng hàm createHeader như bên dưới.



```
C:\Users\84918\Desktop\Đai_Hoc\hk1_nam_2\NetWork\Do_an_socket\server.py - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

server.py x header404.txt x headerok.txt x Server.py x test.py x
30 return '.docx' in path or '.pptx' in path or '.x
31
32 def createHeader(path):
33     filename = ''
34     if '404' not in path: filename = 'headerok.txt'
35     elif '404' in path: filename = 'header404.txt'
36     readHeader = open(filename)
37     header = readHeader.read()
38     readHeader.close()
39     return header
40
41 def response(path, Client):
[Finished in 0.1s]
Line 37, Column 31 Tab Size: 4 Python
```

Nếu bị lỗi 404 thì ta trả về header của 404 ngược lại ta trả về header 200 OK (tức là thành công). Nội dung của 2 file như sau :



The image displays two screenshots of a Sublime Text editor window, showing the editing of HTTP headers in a file named 'headerok.txt'.

**Top Screenshot:** The editor shows the following content:

```
1 HTTP/1.1 200 OK\r\n
2 Content-Encoding: UTF-8\r\n
3
4
```

The status bar at the bottom indicates 'Line 4, Column 1'.

**Bottom Screenshot:** The editor shows the following content:

```
1 HTTP/1.1 404 Not Found\r\n
2 Content-Encoding: UTF-8\r\n
3
4
```

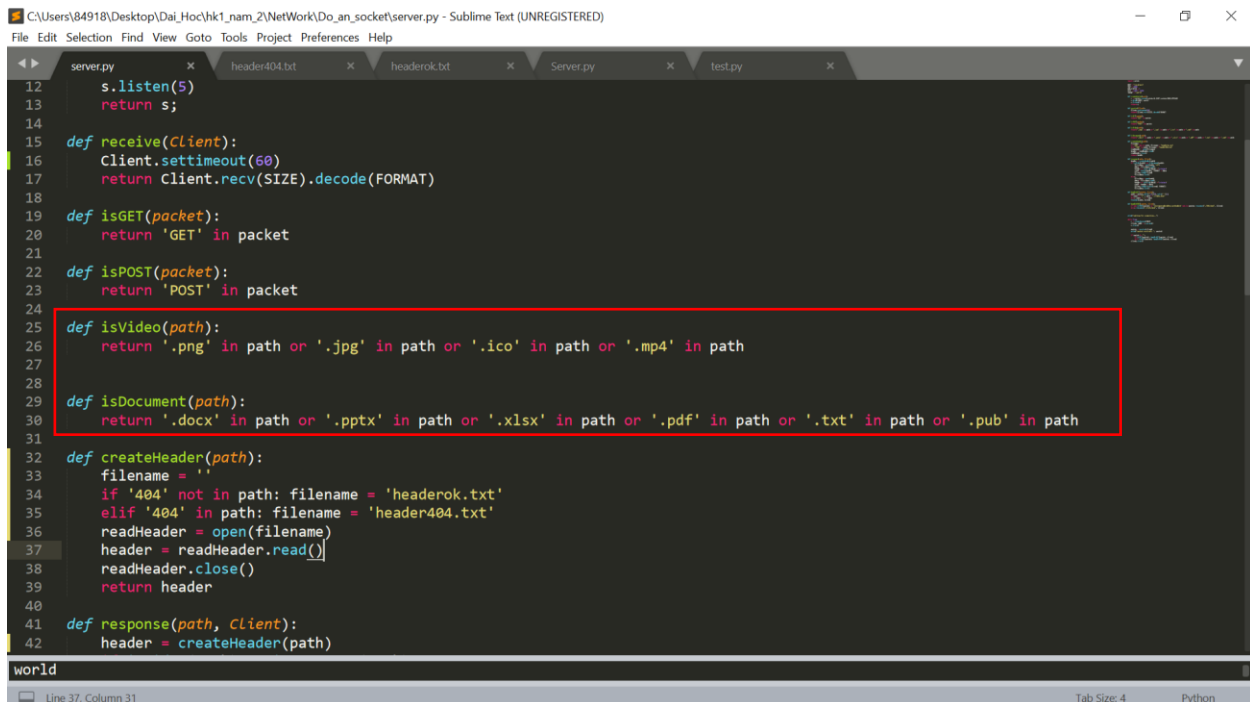
The status bar at the bottom indicates 'Line 4, Column 1'.

Thành Phố Hồ Chí Minh – 2020





Sau khi đã có header ta kiểm tra định dạng file nếu là document hoặc video hay ảnh ta sẽ đọc theo kiểu binary (nhị phân vì các file này được mã hóa). Dưới đây là cách implement hai hàm này:

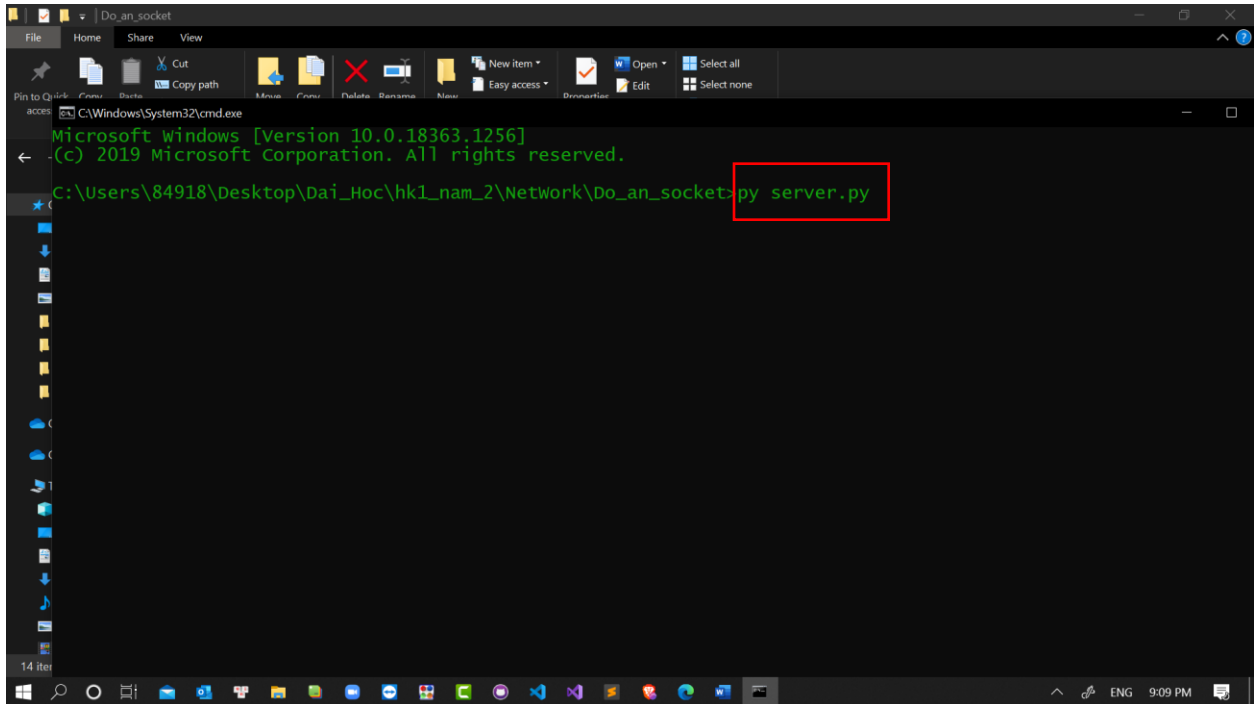


```
server.py
12 s.listen(5)
13 return s;
14
15 def receive(Client):
16     Client.settimeout(60)
17     return Client.recv(SIZE).decode(FORMAT)
18
19 def isGET(packet):
20     return 'GET' in packet
21
22 def isPOST(packet):
23     return 'POST' in packet
24
25 def isVideo(path):
26     return '.png' in path or '.jpg' in path or '.ico' in path or '.mp4' in path
27
28
29 def isDocument(path):
30     return '.docx' in path or '.pptx' in path or '.xlsx' in path or '.pdf' in path or '.txt' in path or '.pub' in path
31
32
33 def createHeader(path):
34     filename = ''
35     if '404' not in path: filename = 'headerok.txt'
36     elif '404' in path: filename = 'header404.txt'
37     readHeader = open(filename)
38     header = readHeader.read()
39     readHeader.close()
40     return header
41
42 def response(path, Client):
43     header = createHeader(path)
```

Ta mở file đó ra đọc nội dung sau đó gửi header + data đó về cho client. Ngược lại nếu không phải các file có định dạng trên ta sẽ mở ra đọc bình thường rồi cũng gửi header + data về cho client xử lý và hiển thị.

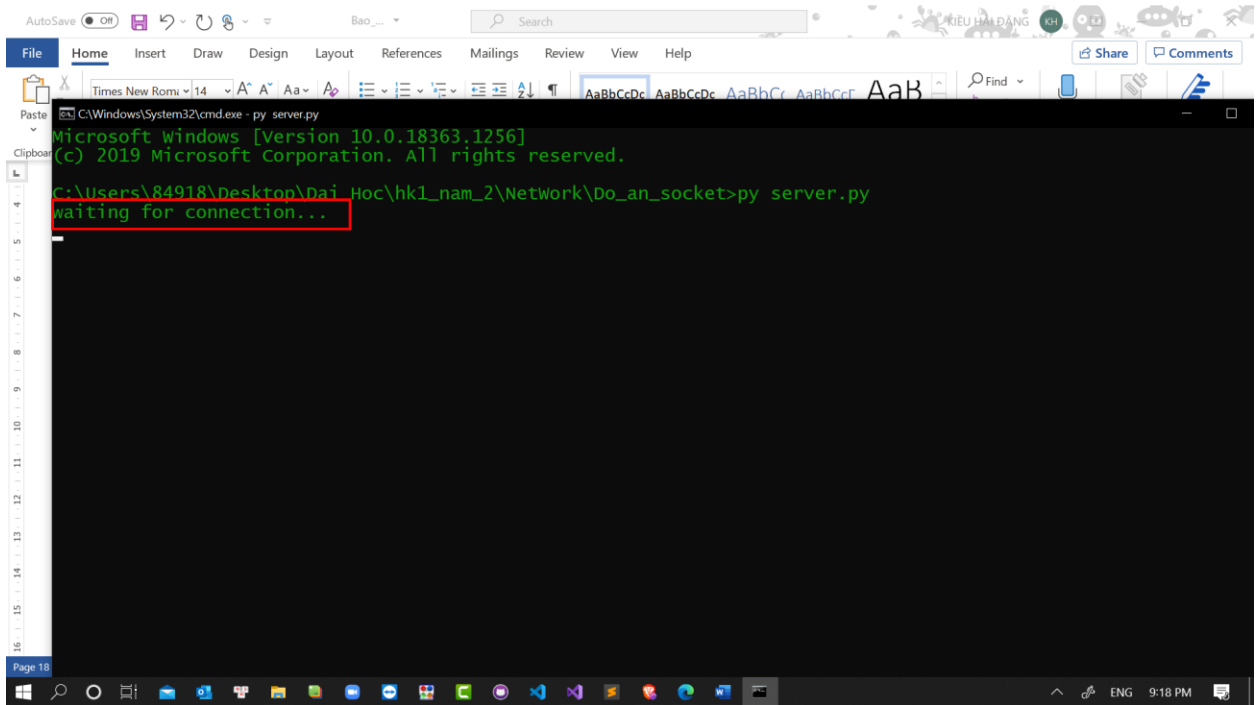


### III. Demo các bước thực hiện:



Đầu tiên ta truy cập vào thư mục chứa file server.py bật cmd lên và gõ lệnh py server.py và server được khởi động như hình bên dưới.





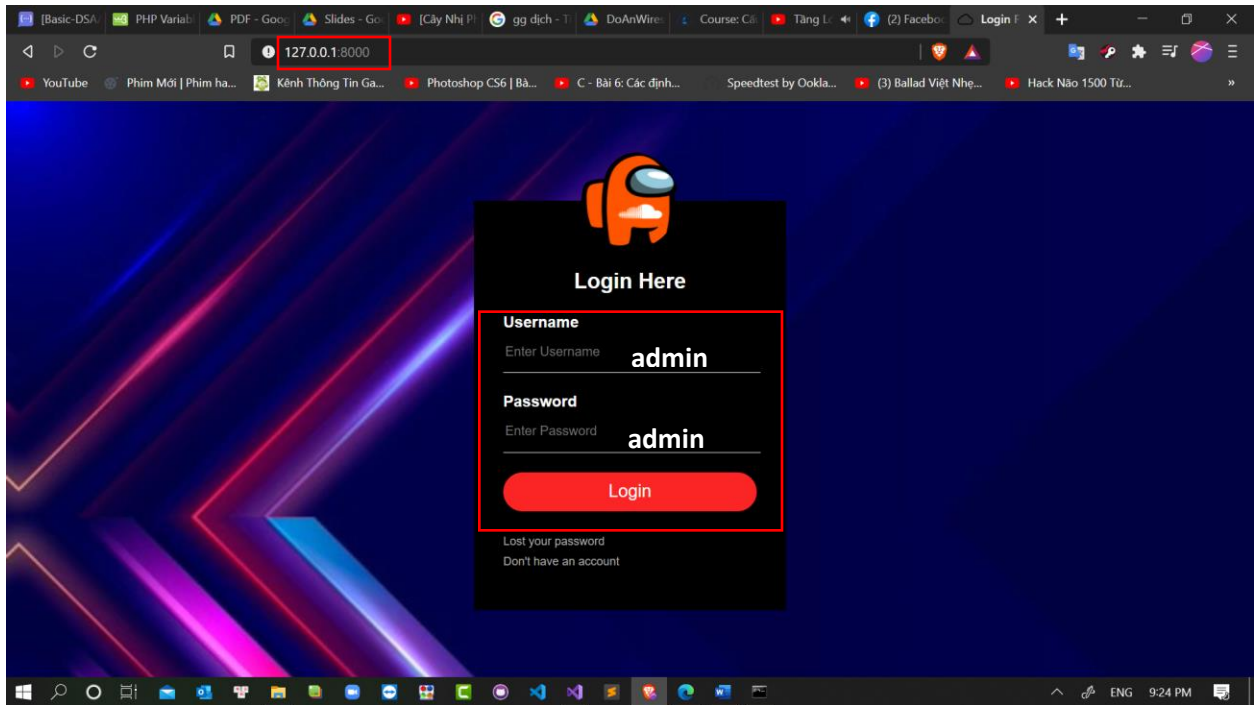
The screenshot shows a Microsoft Word document with a Windows command prompt window open. The command prompt displays the following text:

```
Microsoft Windows [Version 10.0.18363.1256]  
(c) 2019 Microsoft Corporation. All rights reserved.  
  
C:\Users\84918\Desktop\Đai_Hoc\hkl_nam_2\Network\Do_an_socket>py server.py  
waiting for connection...
```

The text "waiting for connection..." is highlighted with a red box. The Word document's ribbon is visible at the top, and the taskbar is at the bottom.

Server đã được khởi động và đang chờ kết nối.

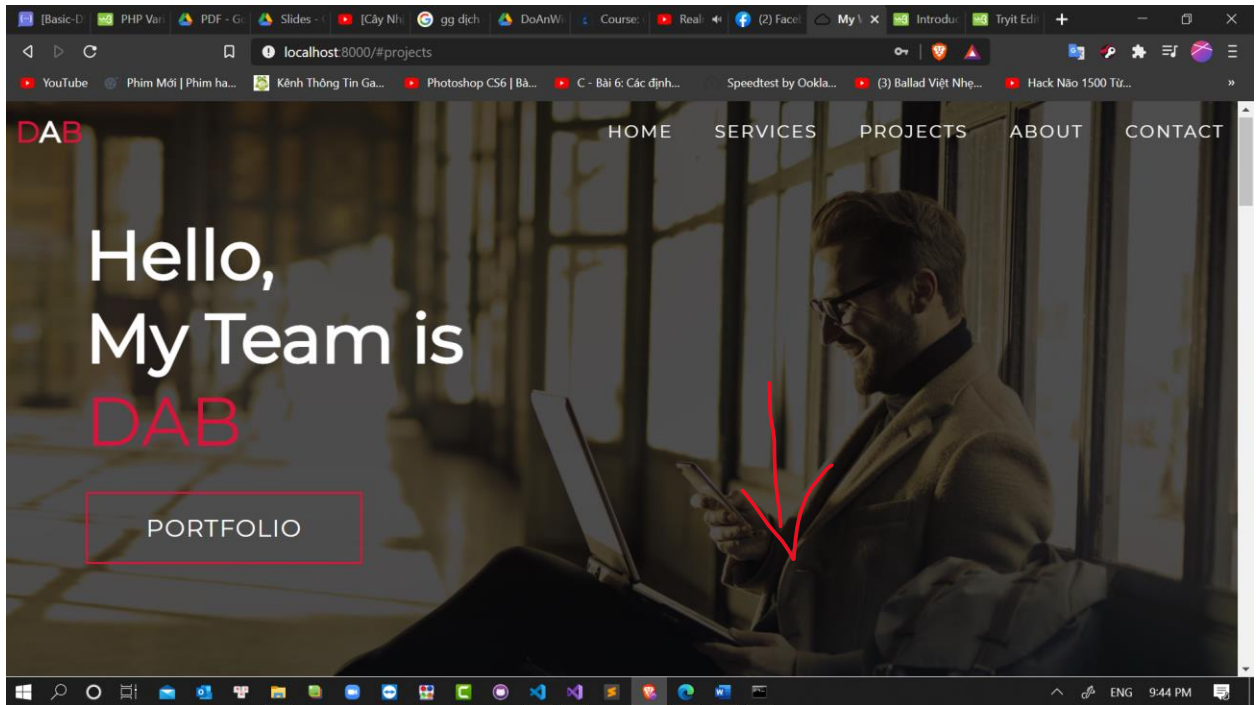




Ở bước tiếp theo này ta bật trình duyệt rồi truy cập theo đường link 127.0.0.1:8000 để có thể truy cập được vào server và ra trang index.

Ở trang index này nếu ta nhập đúng password và username là admin rồi ấn login thì sẽ truy cập được đến trang info ngược lại thì sẽ bị dẫn đến trang 404.

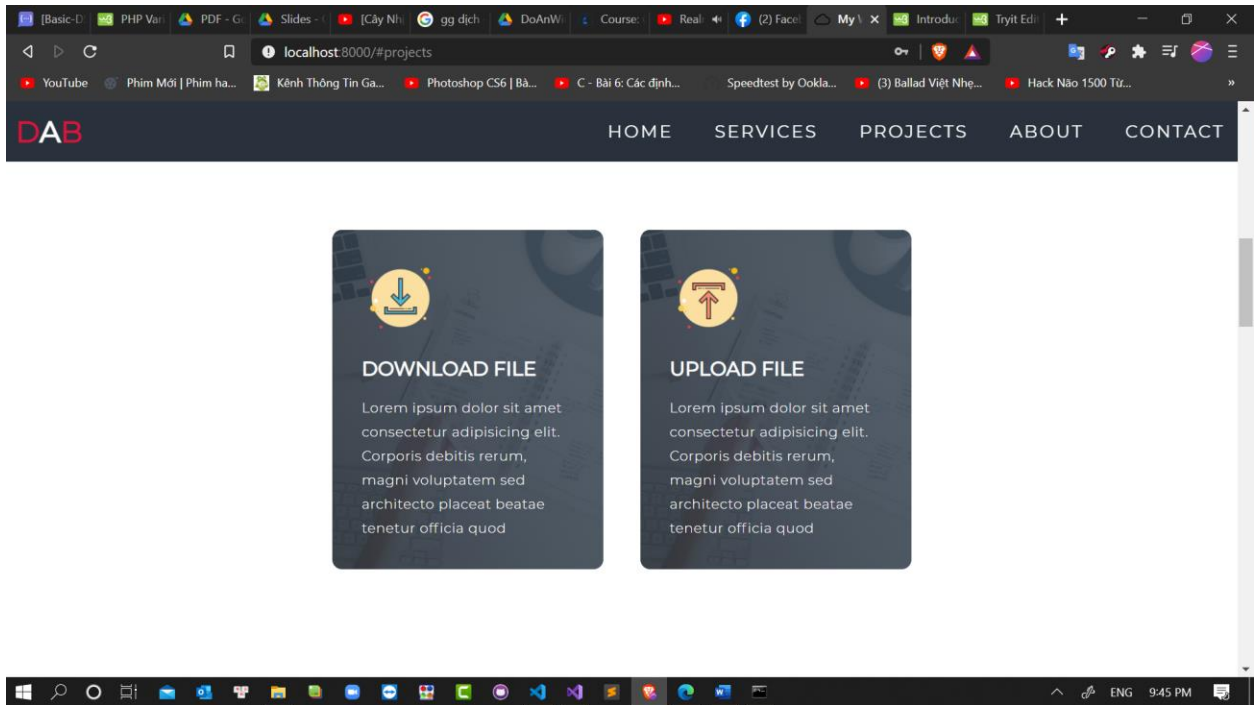




Ở giao diện info này khi ta kéo xuống dưới đầu tiên sẽ gặp nơi để có thể download như hình bên dưới.

Thành Phố Hồ Chí Minh – 2020

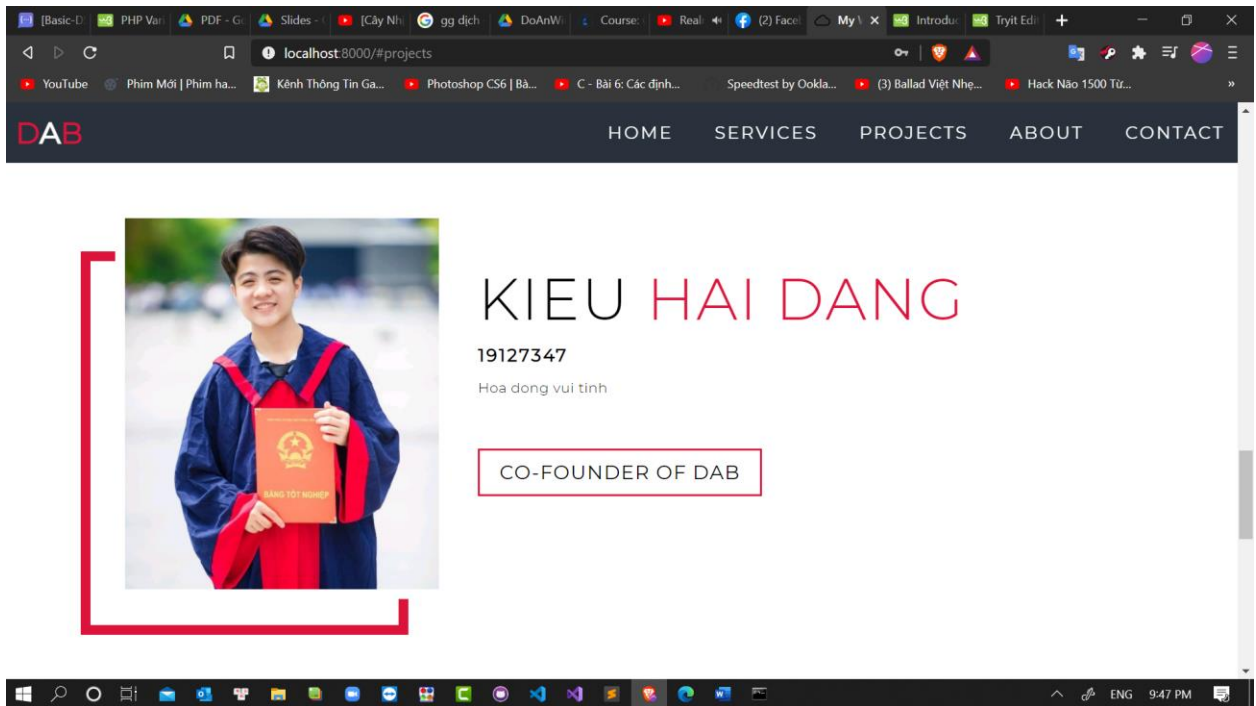
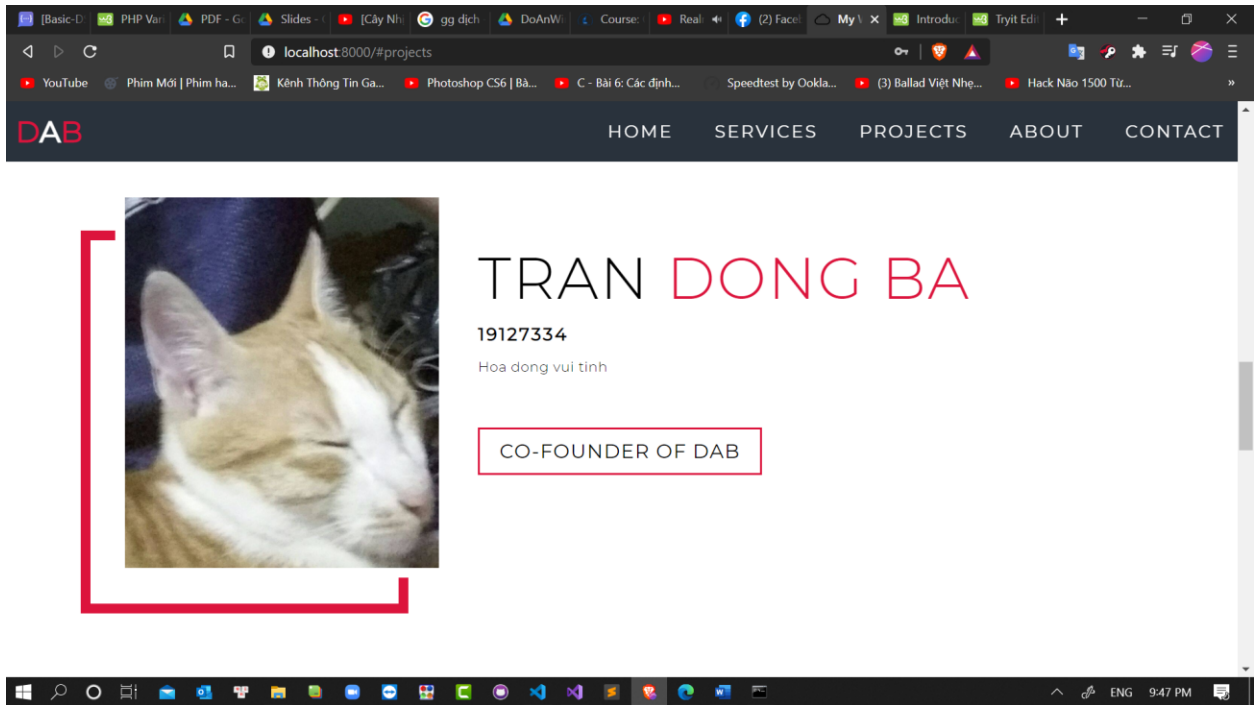




Nếu kéo xuống nữa sẽ là phần giới thiệu về các thành viên của nhóm.

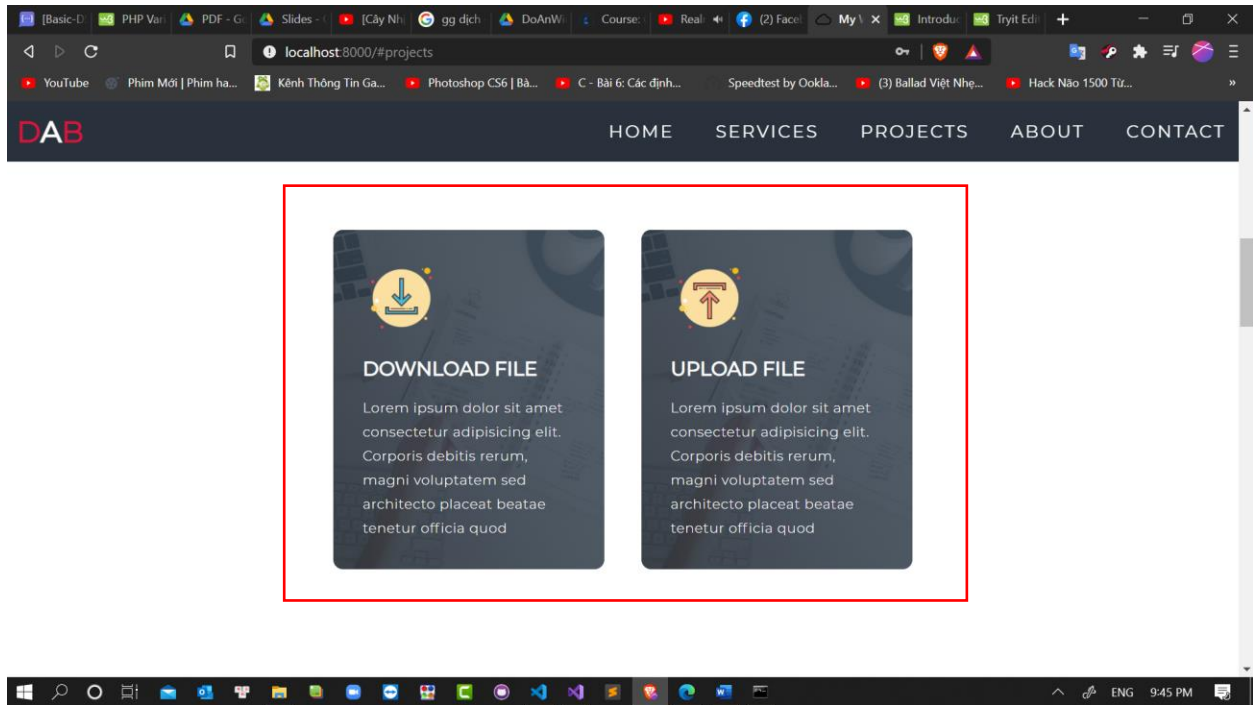
**Thành Phố Hồ Chí Minh – 2020**





Thành Phố Hồ Chí Minh – 2020

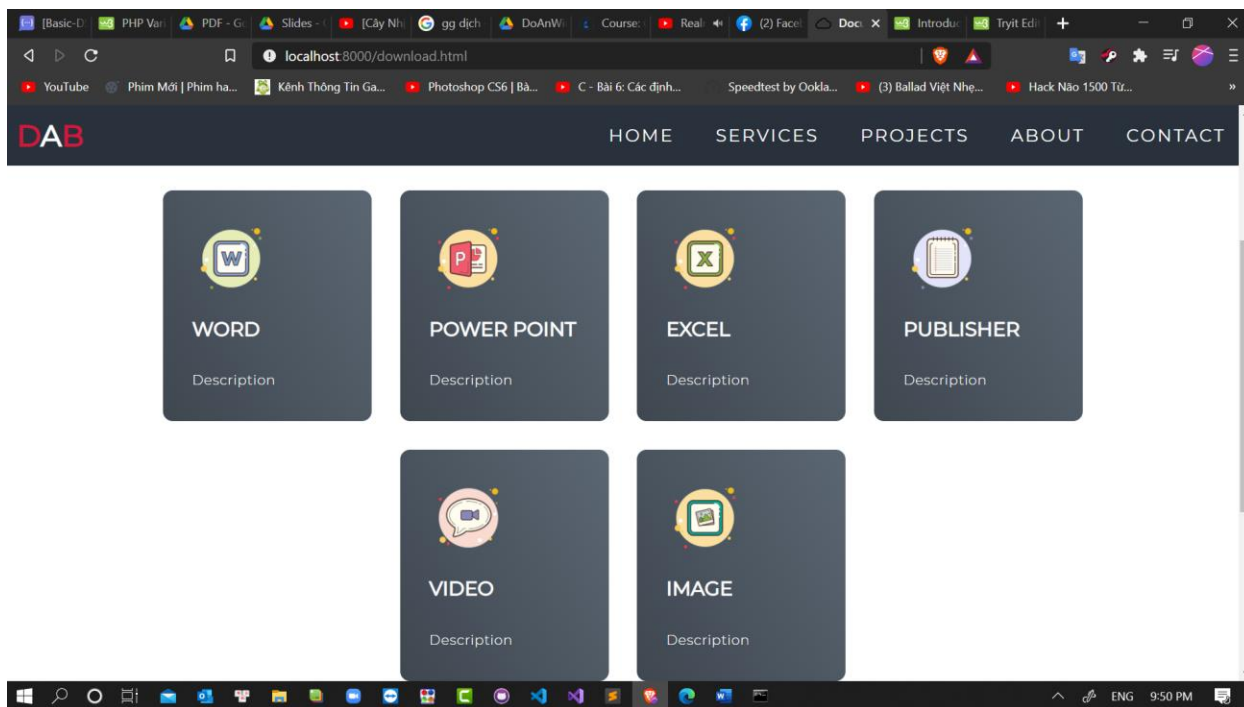




Ở phần download upload này do nhóm chưa cài đặt upload nên ấn vào đâu cũng sẽ ra trang download và ở đây nhóm sẽ demo khi click vào phần download file.



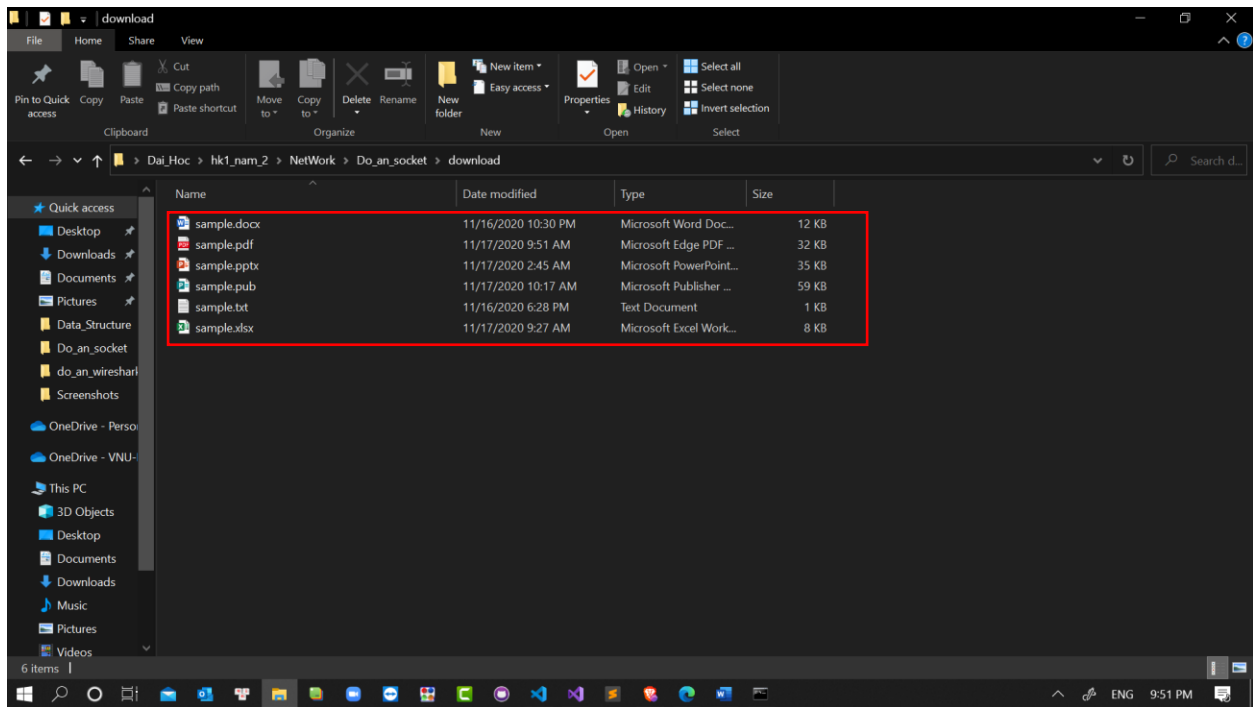




Ở trang download này nhóm có để một số loại file thông dụng và tất cả file này được cất bên trong thư mục chứa server.

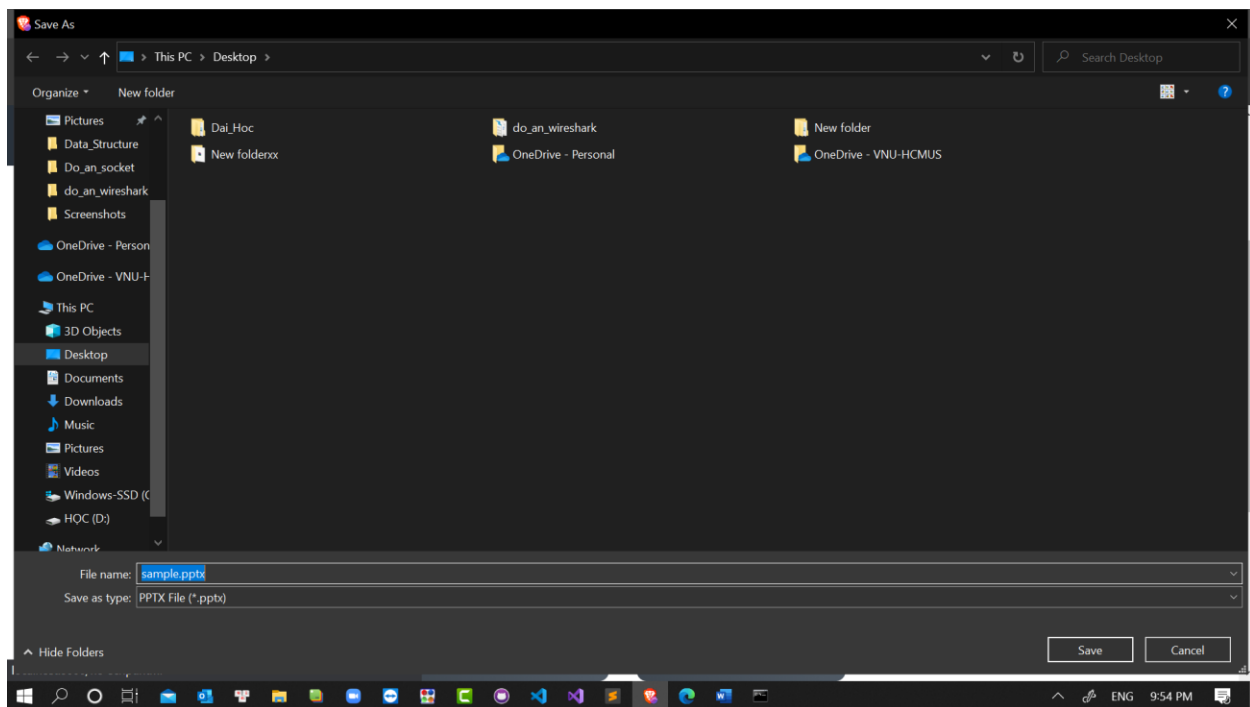
Thành Phố Hồ Chí Minh – 2020





Các file được cất trong thư mục Download và hình bên trên là một số định dạng được hỗ trợ.





Và cuối cùng khi ta ấn vào một button download nào trên trang web thì file sẽ được down về máy.



# HẾT

Thành Phố Hồ Chí Minh – 2020

---

