

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

MÔN HỌC: MẠNG MÁY TÍNH

LỚP: 18_4

BÁO CÁO ĐỒ ÁN 3



Sinh viên thực hiện:

Nguyễn Trần Tổ – 18120093

Bùi Văn Thanh – 18120559

Giảng viên hướng dẫn:

Thầy Lê Hà Minh

I. Thông tin thành viên và phân công công việc

MSSV	Họ tên	Phân công	Tỷ lệ đóng góp
18120093	Nguyễn Trần Tố	<ul style="list-style-type: none">- Xây dựng Web Server- Trang index.html- Làm báo cáo	55%
18120559	Bùi Văn Thanh	<ul style="list-style-type: none">- Xây dựng trang báo lỗi 404.html- Trang info.html	45%

II. Nội dung

a. Môi trường lập trình

- Ngôn ngữ Python 3, html, css
- Mức độ hoàn thành: 100%

Yêu cầu	Mức độ hoàn thành
Xây dựng trang web index.html	100%
Xây dựng trang thông tin thành viên nhóm info.html	100%
Dùng HTTP method là POST để gửi request đăng nhập lên Web Server	100%
Xây dựng trang báo lỗi 404.html	100%

b. Cách thực hiện

- Xây dựng hàm HostServer để tạo socket và gán địa chỉ Server= '127.0.0.1' và POST=8000

```
PORT=8000
SERVER = '127.0.0.1' #PORT và SERVER IP Address
```

```
#Tạo HostServer
def HostServer():
    HOST = socket(AF_INET, SOCK_STREAM) #Tạo socket

    HOST.bind((SERVER, PORT)) #gán IP Address và PORT
```

- Lắng nghe kết nối và tạo vòng lặp, gọi hàm xử lý kết nối, đóng kết nối

```

HOST.listen(1)                                #lắng nghe kết nối
print("Server is listening at http://" + SERVER + ":" + str(PORT))
while True:
    connection, address = HOST.accept()        #chấp nhận kết nối từ Client
    print("Connected to " + str(address))
    Handle_Client_Request(connection)          #gọi hàm xử lý request từ client
    connection.close()                        #đóng kết nối đến Client

```

- Đóng Server

```

print("Server is closing...")
HOST.close()    #Đóng HostServer

```

- Xây dựng hàm Handle_Client_Request để xử lý thông tin requests nhận được

```

#hàm xử lý request nhận được
def Handle_Client_Request(connection):
    request = connection.recv(1024)        #nhận request

```

- Nhận request, sau đó decode thành 'utf - 8', sau đó tách request nhận được để xử lý method

```

#hàm xử lý request nhận được
def Handle_Client_Request(connection):
    req request: bytes n.recv(1024)        #nhận request
    if request:
        request = request.decode('utf-8')    #decode to UTF-8
    else:
        print("We Not received any request\n")    #không nhận được request in thông báo rồi thoát ra
        return

    print("Client request: \n" + request)

    str_list = request.split(' ')    #tách chuỗi để xử lý
    method = str_list[0]            #method là loại request

```

- Xử lý method='GET'
 - + Mở file index.html để đọc nội dung, sau đó đóng file
 - + Tạo response gửi lại client

```

if method == "GET":                #xử lý method GET
    req_file = str_list[1]         #tên file
    req_file = req_file.lstrip('/')
    req_file = req_file.replace("%20", " ")
    if (req_file == ''):           #nếu không có tên file thì gán là index.html
        req_file = 'index.html'

    file = open(req_file, 'rb')    #mở file
    response = file.read()         #đọc nội dung file vào response
    file.close()
    header = "HTTP/1.1 200 OK\n"   #thêm header của response
    if req_file.endswith(".html"):
        header += "Content-Type: text/html\n\n"
    elif req_file.endswith(".css"):
        header += "Content-Type: text/css\n\n"
    else:
        header += "Content-Type: */*\n\n"

    final_res = header.encode('utf-8')
    final_res += response
    connection.send(final_res)     #gửi response cho Client

```

- Xử lý method='POST'
- + Tách request nhận được để lấy username và password
- + Kiểm tra thông tin, nếu đúng thì vào trang info.html, sai thì vào trang 404.html

```

elif method == "POST":            #xử lý method POST
    str_split = request.split('\n') #tách chuỗi lấy username và password
    loginInfo = str_split[-1]
    username = loginInfo.split('&')[0]
    password = loginInfo.split('&')[1]
    username = username[9:]
    password = password[9:]
    # password: str
    if username == "admin" and password == "admin": # kiểm tra password, đúng -> truy cập trang info
        header = "HTTP/1.1 301 Moved Permanently\nLocation: /info.html\n" #header của HTTP Redirection
        connection.send(header.encode('utf-8'))
    else: #password sai -> truy cập trang báo lỗi 404
        header = "HTTP/1.1 404 Not Found\n\n"
        file = open("404.html", "rb")
        response = file.read()
        file.close()
        final_res = header.encode('utf-8')
        final_res += response
        connection.send(final_res)

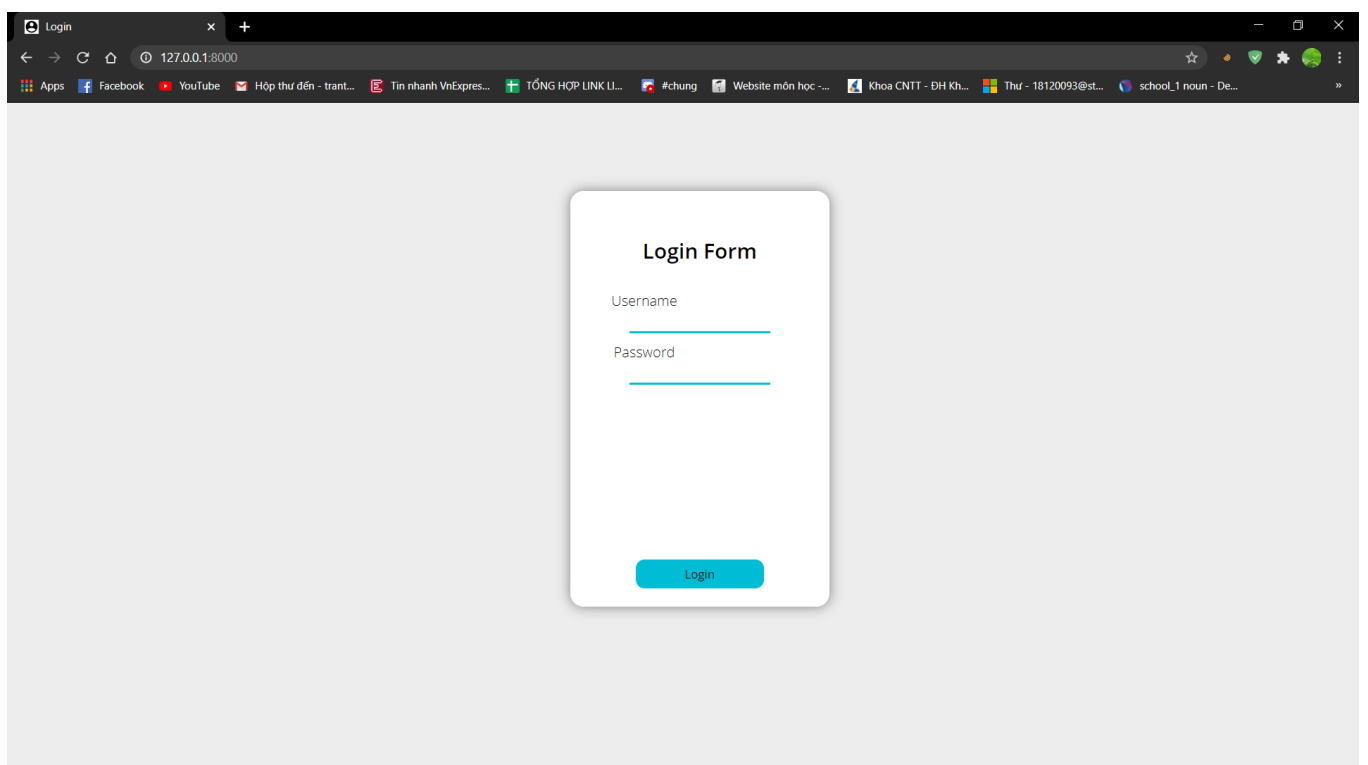
```

c. Cách test ứng dụng

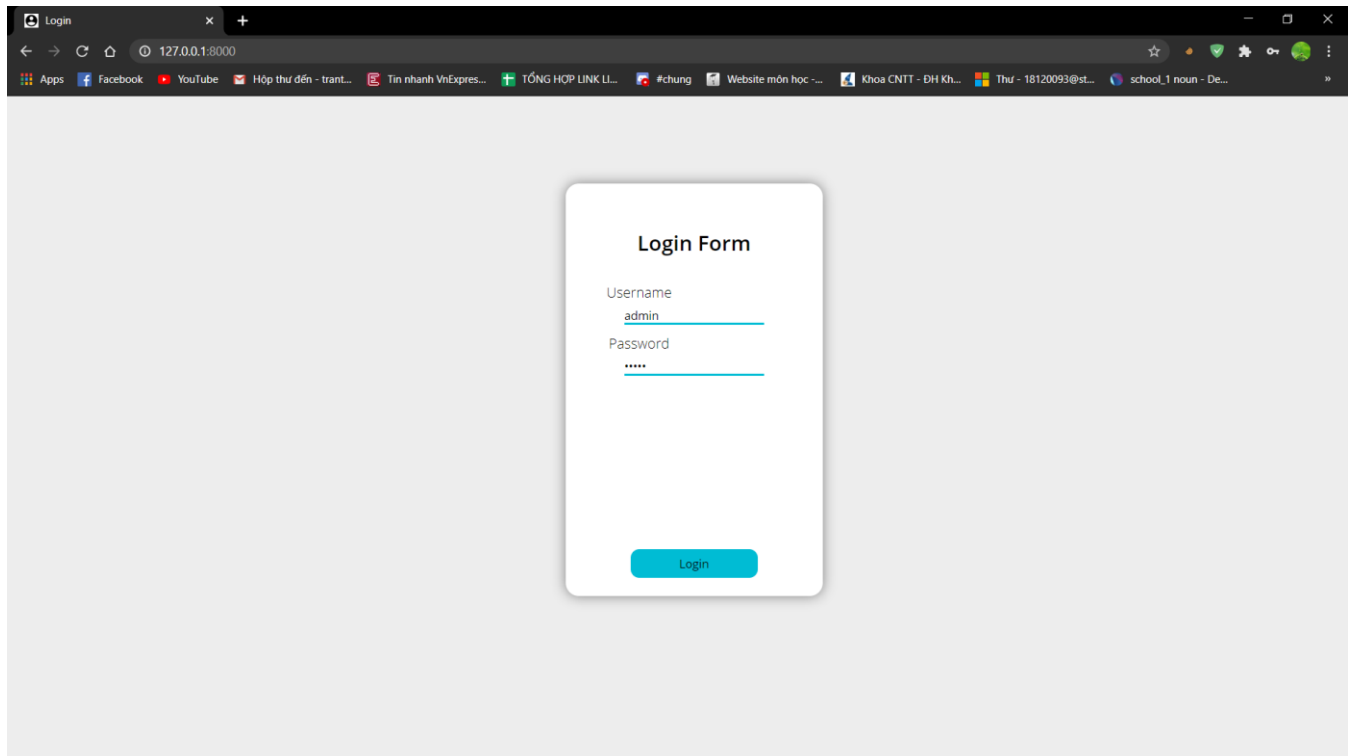
- Chạy file 18120093_18120559.py sau đó vào google truy cập đường dẫn 127.0.0.1:8000

The screenshot shows the Visual Studio Code interface with a Python file named `18120093_18120059.py` open. The code is a simple web server using the `socket` module. It listens on `127.0.0.1:8000` and handles GET requests by serving the `index.html` file. The terminal at the bottom shows the command prompt running the script, and the output indicates the server is listening at `http://127.0.0.1:8000`.

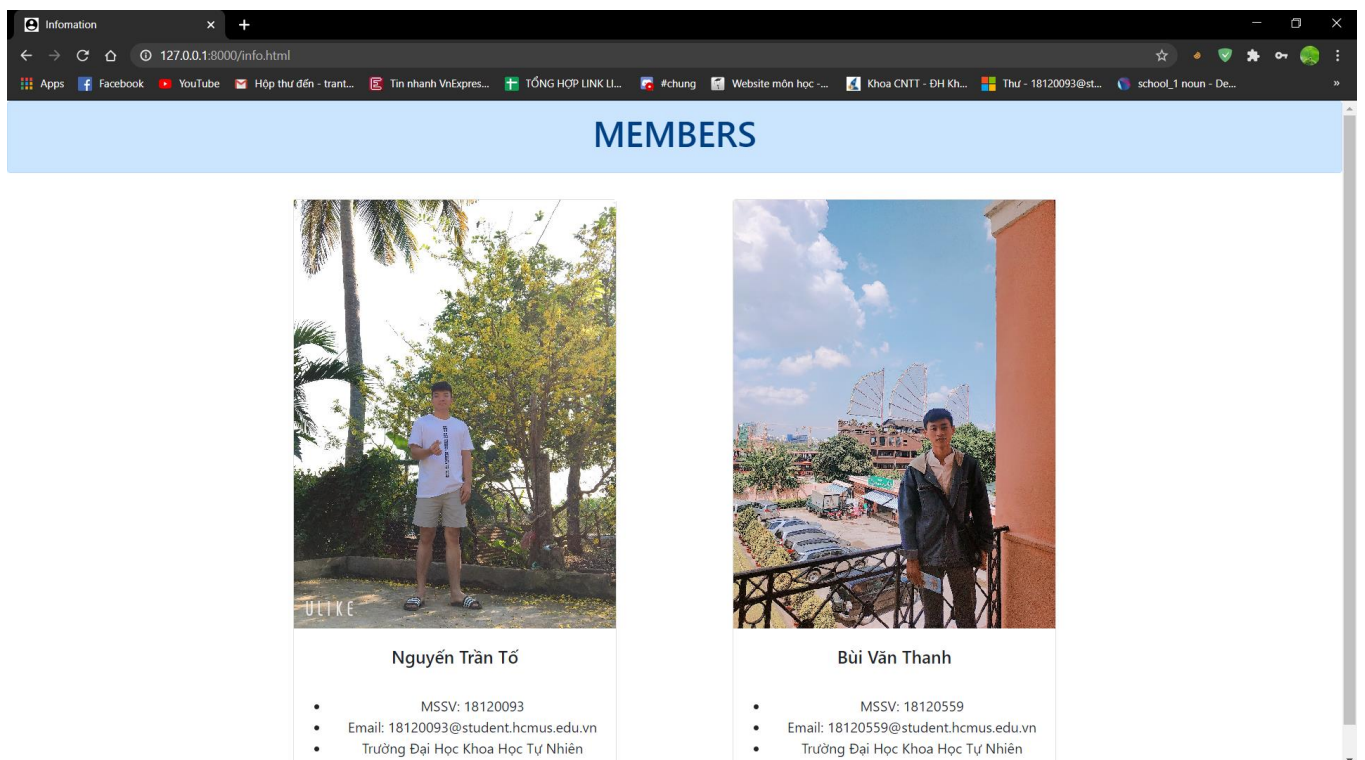
```
1 from socket import AF_INET, socket, SOCK_STREAM
2
3
4 PORT=8000
5 SERVER = '127.0.0.1' #PORT và SERVER IP Address
6
7 #hàm xử lý request nhận được
8 def Handle_Client_Request(connection):
9     request = connection.recv(1024) #nhận request
10    if request:
11        request = request.decode('utf-8') #decode to UTF-8
12    else:
13        print("We Not received any request\n") #không nhận được request in thông báo rồi thoát ra
14        return
15
16    print("Client request: \n" + request)
17
18    str_list = request.split(' ') #tách chuỗi để xử lý
19    method = str_list[0] #method là loại request
20
21    if method == "GET": #xử lý method GET
22        req_file = str_list[1] #tên file
23        req_file = req_file.lstrip('/')
24        req_file = req_file.replace("%20", " ")
25        if (req_file == ''): #nếu không có tên file thì gán là index.html
26            req_file = 'index.html'
27
28        file = open(req_file, 'rb') #mở file
29        response = file.read() #đọc nội dung file vào response
30        file.close()
31        header = "HTTP/1.1 200 OK\n" #thêm header của response
32        if req_file.endswith(".html"):
```



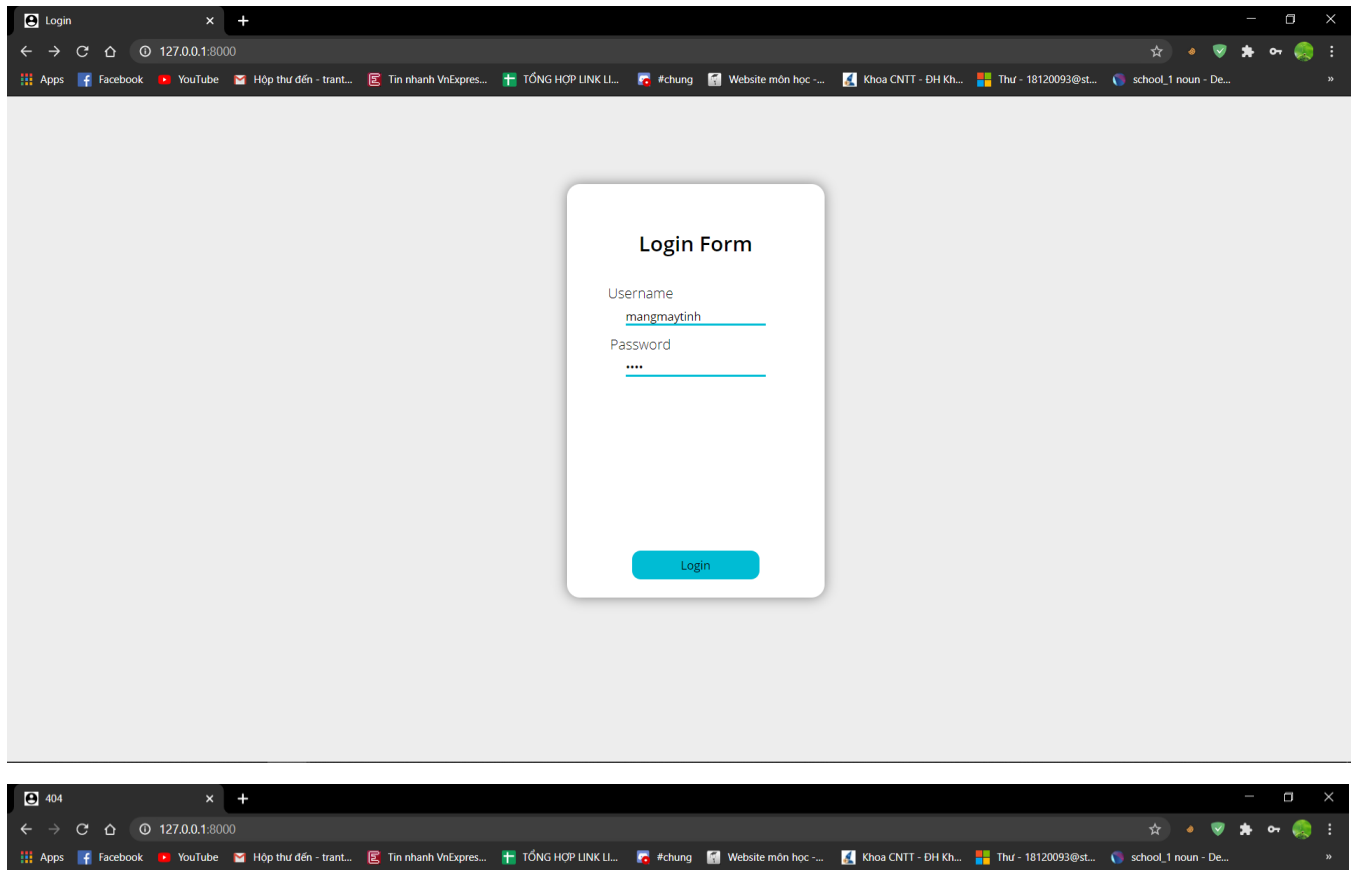
- Nhập username và password



- Nếu đúng sẽ dẫn đến trang info.html



- Nếu sai sẽ dẫn đến trang 404.html



404

Page not found

III. Tài liệu tham khảo:

- Tài liệu socket Thầy Lê Hà Minh

<https://drive.google.com/file/d/10lBDAwpoKDvZSqYgrrjKAqupyov44pEJ/view>

- <https://topdev.vn/blog/huong-dan-cao-mot-http-server-bang-python/>

- https://www.youtube.com/watch?v=CxMbQ2AqteE&list=PLLAJJPGNwNkgHoNSB9xq22EJ_Z1rX7Ygs
- <https://toidicode.com/lap-trinh-mang-voi-module-socket-trong-python-364.html>