



**fit@hcmus**

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

# **ĐỒ ÁN THỰC HÀNH** **MẠNG MÁY TÍNH**

## **PROJECT-01: SOCKET PROGRAMMING**

*GV hướng dẫn: Cô Chung Thùy Linh*

Thành viên

Lương Ánh Nguyệt – 19120315


Đoàn Thu Ngân - 19120302

Thành Phố Hồ Chí Minh, ngày 20 tháng 12 năm 2020

# MỤC LỤC

I. THÔNG TIN ĐỒ ÁN .....	2
II. THỰC HIỆN ỨNG DỤNG.....	2
1. Xác định tổng quát các thông tin server .....	2
2. Xây dựng các hàm cơ bản.....	3
III. KIỂM THỬ ỨNG DỤNG .....	7
IV. DANH MỤC TÀI LIỆU THAM KHẢO .....	11

# I. THÔNG TIN ĐỒ ÁN

 **Ý tưởng chính:** Viết ứng dụng Web Server đơn giản bằng cách sử dụng lập trình socket để trả về nội dung trang web sau:

✓ **Trang index.html:** Trang để người dùng đăng nhập. Ở trang này, người dùng nhập Username là "*admin*" và Password là "*admin*" để đăng nhập (Sử dụng phương thức HTTP là POST để gửi thông tin người dùng nhập đến Web Server).

✓ **Trang info.html:** Nếu thông tin đăng nhập chính xác (*admin* / *admin*), server sẽ chuyển hướng đến trang info.html. Trang này chứa hình ảnh và thông tin về các thành viên trong nhóm.

✓ **Trang 404.html:** Nếu thông tin đăng nhập sai, thì lỗi 404 được trả về trình duyệt.

✓ **Trang file.html:** Hiện thị danh sách các tệp cho phép người dùng tải xuống các tệp này thông qua trình duyệt Web. (Loại tệp bao gồm: video, văn bản, âm thanh, hình ảnh)

## II. THỰC HIỆN ỨNG DỤNG

### 1. Xác định tổng quát các thông tin server

**Xác định kiến trúc mạng:** Client – Server

**Giao thức sử dụng tầng transport:** TCP (Transmission Control Protocol)

Sử dụng TCP, các ứng dụng trên các máy chủ được nối mạng có thể tạo các "kết nối" với nhau, mà qua đó chúng có thể trao đổi dữ liệu hoặc các gói tin. Giao thức này đảm bảo chuyển giao dữ liệu tới nơi nhận một cách đáng tin cậy và đúng thứ tự.

**Port sử dụng cho Server:** 7000

## 2. Xây dựng các hàm cơ bản

- **Khởi tạo Webserver:** Web Server sử dụng port 7000 để kết nối với các client.

```
HOST = ''
PORT = 7000
ADDR = (HOST, PORT)
CHUNK_SIZE = 65536

SERVER = socket(AF_INET, SOCK_STREAM)
SERVER.bind(ADDR)

if __name__ == "__main__":
    try:
        SERVER.listen(5)
        print("Waiting for Client...")

        ACCEPT_THREAD = Thread(target=WaitingConnection)
        ACCEPT_THREAD.start()
        ACCEPT_THREAD.join()
    except:
        print("Error occurred!")
    finally:
        SERVER.close()
```

- **Hàm WaitingConnection():** Server chờ kết nối từ các client

```
def WaitingConnection():
    while True:
        Client, Address = SERVER.accept()
        print("Client", Address, "connected!")
        Thread(target=TakeRequest, args=(Client,)).start()
```

- **Hàm ReadRequest(Client):** Đọc yêu cầu từ client đã kết nối

```
def ReadRequest(Client):
    request = ""
    Client.settimeout(1)
    try:
        request = Client.recv(1024).decode()
        while (request):
            request = request + Client.recv(1024).decode()
    except socket.timeout: # fail after 1 second of no activity
        if not request:
            print("Didn't receive data! [Timeout]")
    finally:
        return request
```

- **Hàm TakeRequest(Client):** Xử lý, phân tích yêu cầu từ client rồi từ đó gọi hàm thích hợp để trả về các response tương ứng cho client

```
def TakeRequest(Client):
    while True:
        Request = ReadRequest(Client)
        if not Request:
            Client.close()
            break
        print("--> Got a request")
        if "GET / HTTP/1.1" in Request or "GET /index" in Request:      #Enter localhost:[port] or localhost:[port]/index
            SendHomepage(Client)
        elif "GET /file.html" in Request:      #Click on button linked to download.html
            SendDownloadpage(Client)
        elif "GET /info.html" in Request:
            SendInfo(Client)
        elif "POST" in Request:      #POST request - check username & password
            PostRequest(Client, Request)
        elif "GET /download/" in Request:      #Click download file
            GetFileType(Client, Request)
        elif "GET /404.html" in Request:
            Send404(Client)
```

- **Hàm SendHomepage(Client):** Gửi "*index.html*" đến client (Trang chủ)

```
def SendHomepage(Client):      #Send "index.html" to client (homepage)
    f = open("index.html", "rb")
    L = f.read()
    header = """HTTP/1.1 200 OK
Content-Length: %d

""" % len(L)
    header += L.decode()
    Client.send(bytes(header, "utf-8"))
```

- **Hàm SendDownloadpage(Client):** Gửi "*file.html*" đến client (Trang hiển thị các file để download)

```
def SendDownloadpage(Client):
    f = open("file.html", "rb")
    L = f.read()
    header = """HTTP/1.1 200 OK
Content-Length: %d

""" % len(L)
    header += L.decode()
    Client.send(bytes(header, 'utf-8'))
```

- **Hàm SendInfo(Client):** Gửi “*info.html*” đến client (trang Thông tin thành viên nhóm).

```
def SendInfo(Client):
    #Send "info.html" to client
    f = open ("info.html", "rb")
    L = f.read()
    header = """HTTP/1.1 200 OK
Content-Length: %d

""" % len(L)
    header += L.decode()
    Client.send(bytes(header, "utf-8"))
```

- **Hàm Send404(Client):** Gửi “*404.html*” đến client (trang thông báo 404 Not Found)

```
def Send404(Client):
    f = open ("404.html", "rb")
    L = f.read()
    header = """HTTP/1.1 404 Not Found
Content-Length: %d

""" % len(L)
    header += L.decode()
    Client.send(bytes(header, 'utf-8'))
```

- **Hàm MovetoInfo(Client):** Chuyển hướng đến trang “*info.html*”

```
def MovetoInfo(Client):
    header = """HTTP/1.1 301 Moved Permanently
Location: info.html

"""
    Client.send(bytes(header, 'utf-8'))
```

- **Hàm Moveto404(Client):** Chuyển hướng đến trang “*404.html*”

```
def Moveto404(Client):
    header = """HTTP/1.1 301 Moved Permanently
Location: 404.html

"""
    Client.send(bytes(header, 'utf-8'))
```

- **Hàm PostRequest(Client, Request):** Xử lý yêu cầu dạng POST (kiểm tra username và password)

```
def PostRequest(Client, Request):
    #Handle POST request (check user and pass from client)
    if "username=admin&password=admin" in Request:
        MovetoInfo(Client)
    else:
        Moveto404(Client)
```

- **Hàm SendDownloadFile(Client,filename):** Gửi file client muốn download

```
def SendDownloadFile(Client,filename):
    f=open(filename,"rb")
    header="""HTTP/1.1 200 OK
    Transfer-Encoding: chunked

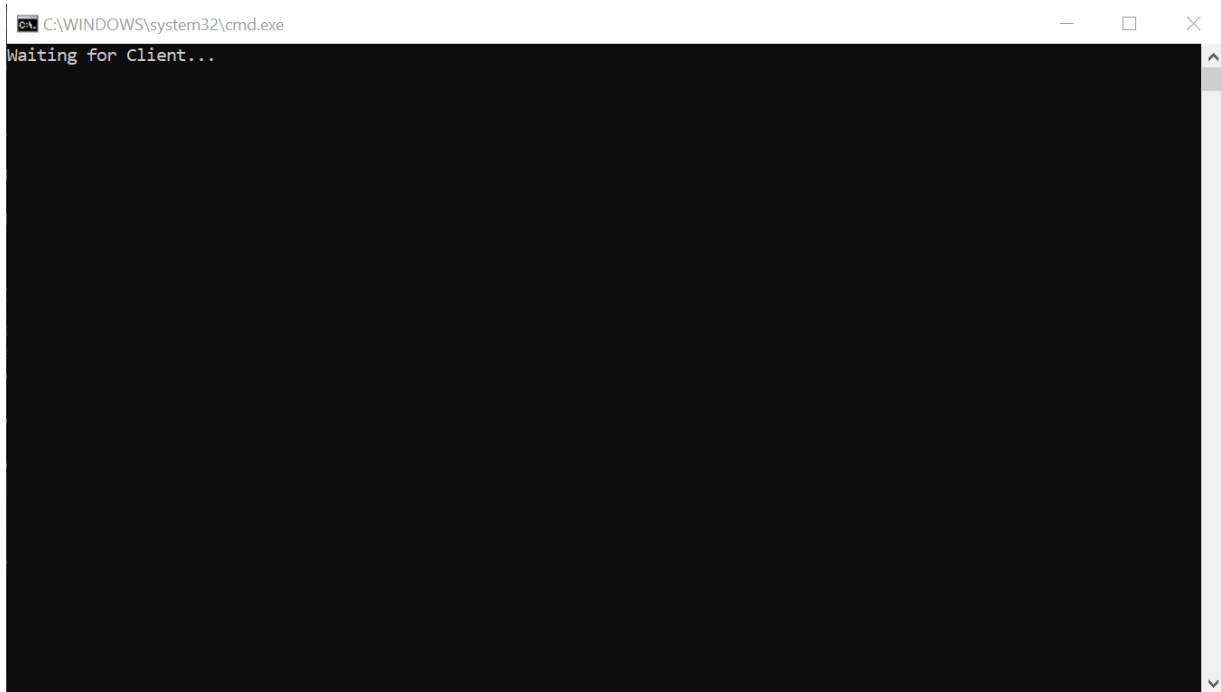
    """
    body = "".encode()
    data = f.read(CHUNK_SIZE)
    while(data):
        body += ("{:x}\r\n".format(len(data))).encode('utf-8')
        body += data
        body += "\r\n".encode()
        data = f.read(CHUNK_SIZE)
    body += "\0\r\n\r\n".encode('utf-8')
    header = header.encode('utf-8') + body
    Client.send(header)
    f.close()
    print("Completed!")
```

- **Hàm GetFileType(Client, Request):** Xử lý request để xác định file client muốn tải

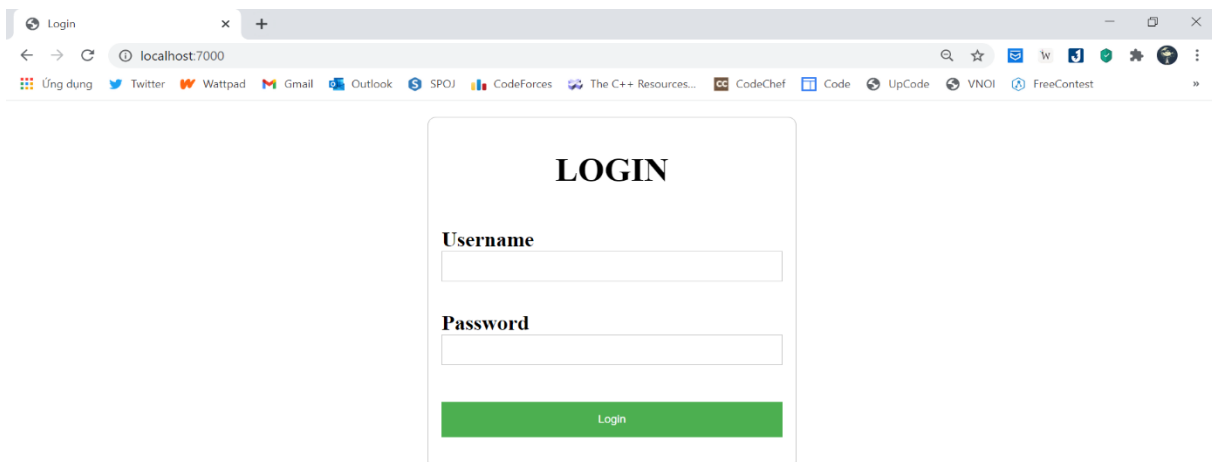
```
def GetFileType(Client, Request):
    kind = ["picture.jpg","video.mp4","text.txt","music.mp3"]
    for x in kind:
        if x in Request:
            print(x, "downloading...")
            x = "download/" + x
            SendDownloadFile(Client,x)
    return
```

### III. KIỂM THỬ ỨNG DỤNG

**Bước 1:** Chạy chương trình (file *Server.py*). Lúc này màn hình console sẽ hiển thị Waiting for Client... , server đã sẵn sàng nhận kết nối từ client

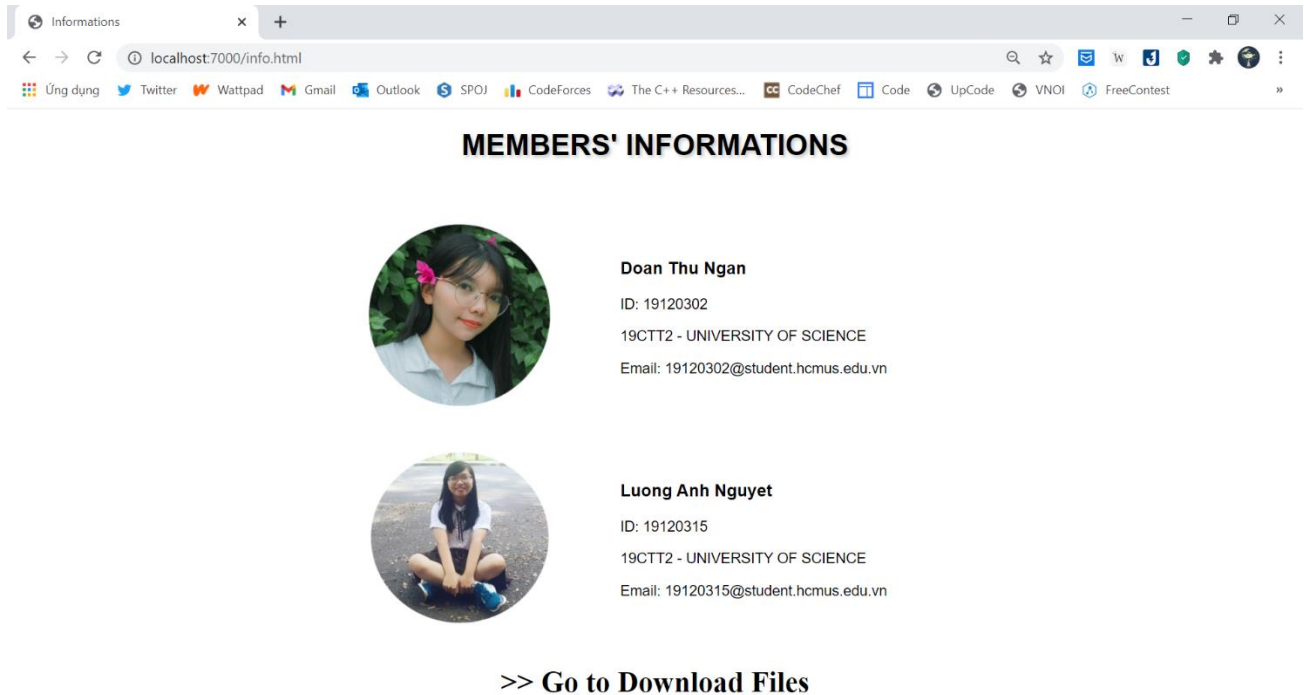


**Bước 2:** Mở trình duyệt web, nhập “localhost:7000” lên thanh công cụ tìm kiếm. Trang Login được hiển thị, người dùng cần nhập Username và Password.





✓ Nhập đúng Username là “*admin*”, Password là “*admin*”, người dùng sẽ được chuyển hướng đến trang “*info.html*” hiển thị thông tin các thành viên trong nhóm.




Information


localhost:7000/info.html

Ứng dụng Twitter Wattpad Gmail Outlook SPOJ CodeForces The C++ Resources... CodeChef Code UpCode VNOI FreeContest

## MEMBERS' INFORMATION



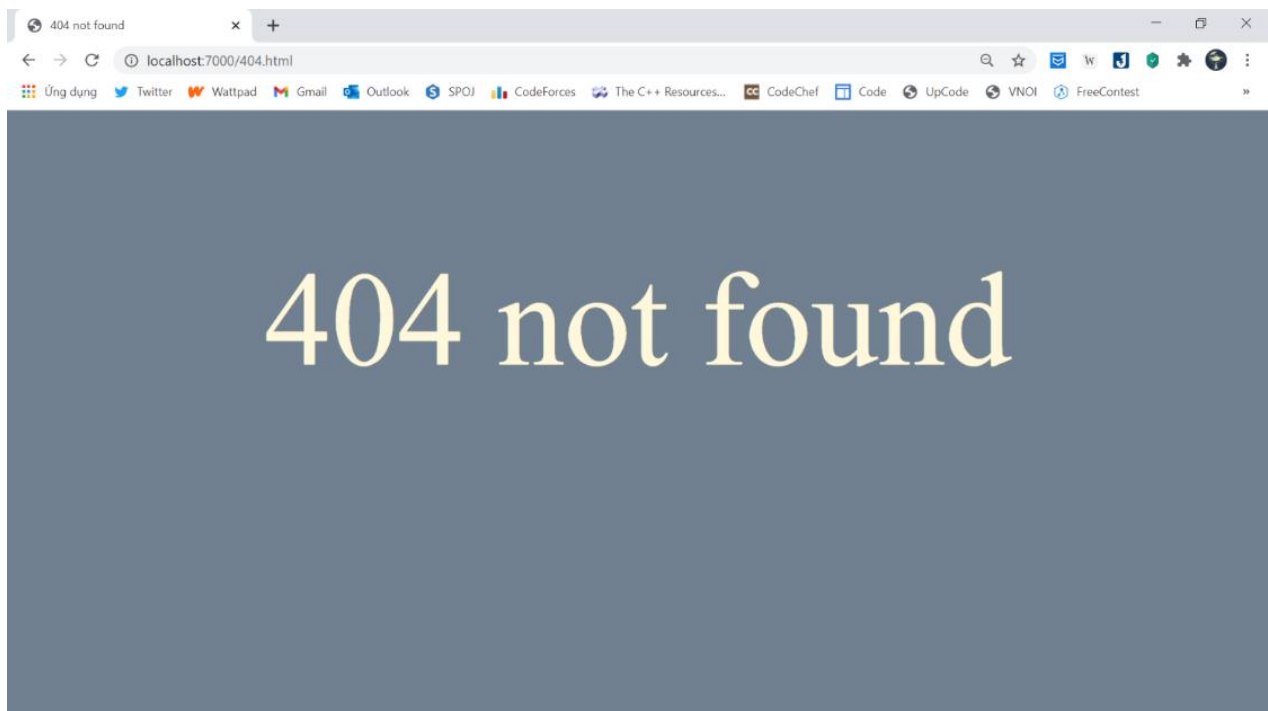
**Doan Thu Ngan**  
ID: 19120302  
19CTT2 - UNIVERSITY OF SCIENCE  
Email: 19120302@student.hcmus.edu.vn



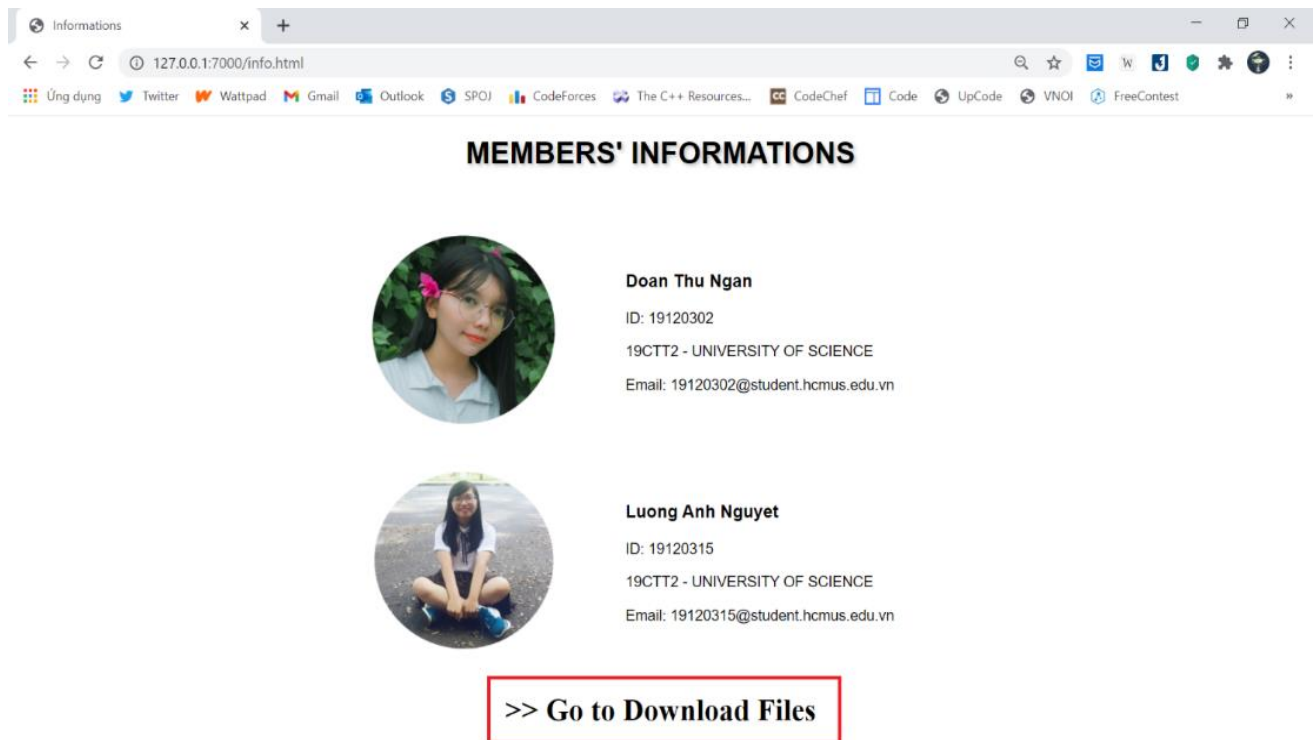
**Luong Anh Nguyet**  
ID: 19120315  
19CTT2 - UNIVERSITY OF SCIENCE  
Email: 19120315@student.hcmus.edu.vn

>> [Go to Download Files](#)


✗ Nếu nhập sai, trang “*404 not found*” sẽ hiện ra




**Bước 3:** Ở trang thông tin thành viên, ta nhấn vào “*Go to Download Files*” để đến trang “*Download Files*”

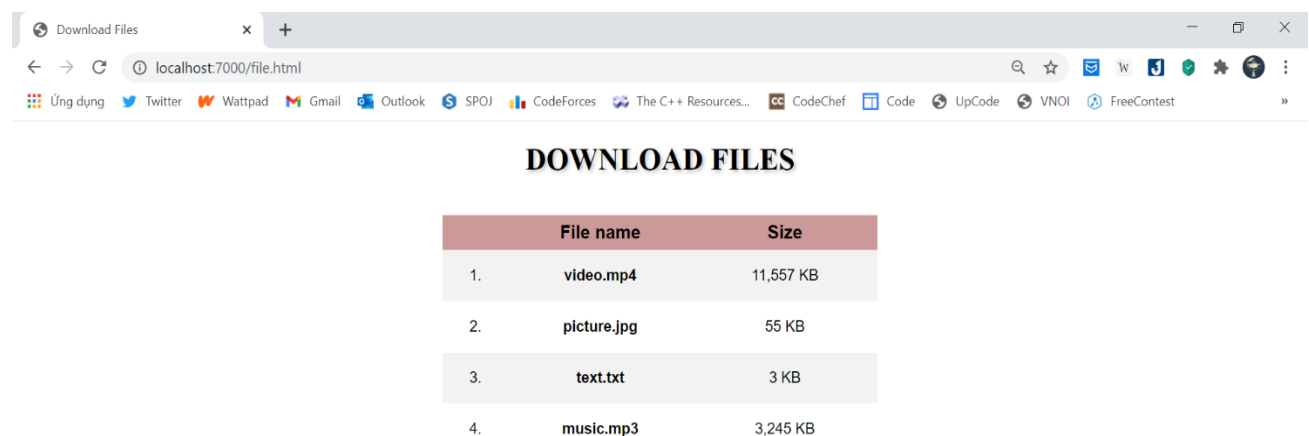


**MEMBERS' INFORMATIONS**

 **Doan Thu Ngan**  
ID: 19120302  
19CTT2 - UNIVERSITY OF SCIENCE  
Email: 19120302@student.hcmus.edu.vn

 **Luong Anh Nguyet**  
ID: 19120315  
19CTT2 - UNIVERSITY OF SCIENCE  
Email: 19120315@student.hcmus.edu.vn

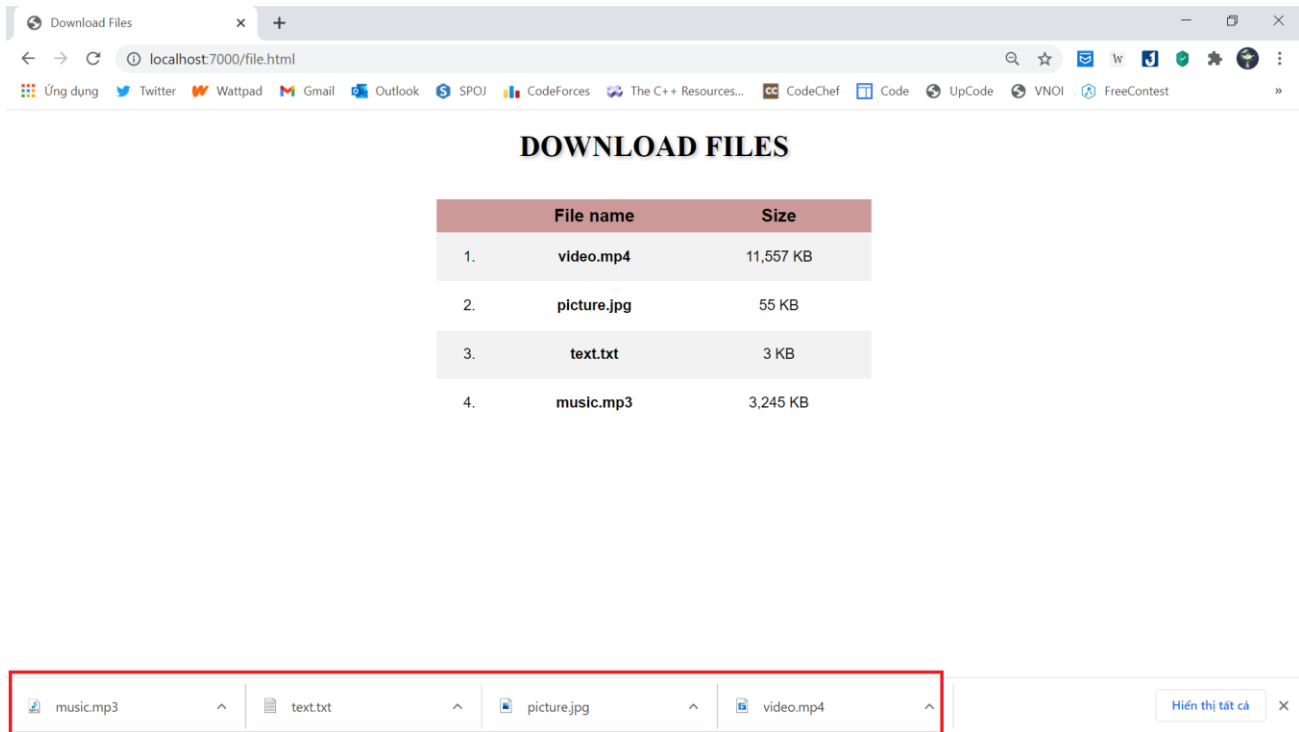
**>> Go to Download Files**



**DOWNLOAD FILES**

	File name	Size
1.	video.mp4	11,557 KB
2.	picture.jpg	55 KB
3.	text.txt	3 KB
4.	music.mp3	3,245 KB

**Bước 4:** Ở trang “*Download Files*”, click chọn vào file bạn muốn tải về

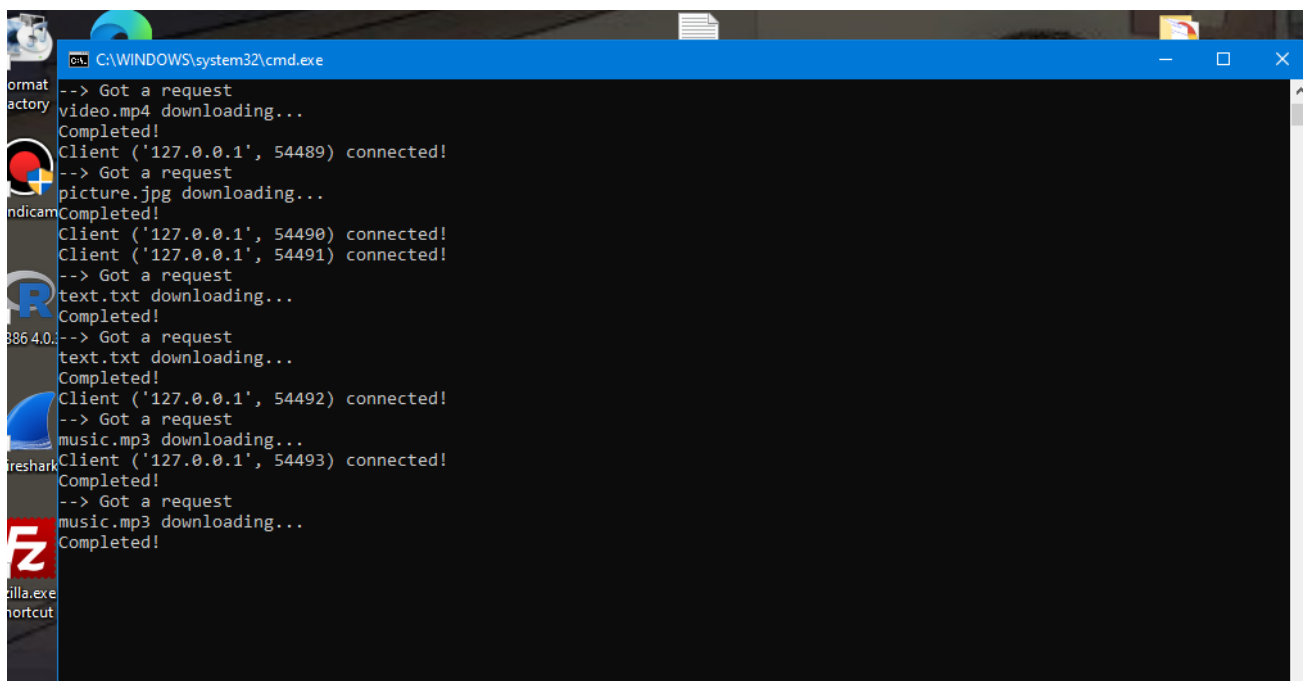


**DOWNLOAD FILES**

	File name	Size
1.	video.mp4	11,557 KB
2.	picture.jpg	55 KB
3.	text.txt	3 KB
4.	music.mp3	3,245 KB

music.mp3    text.txt    picture.jpg    video.mp4    [Hiển thị tất cả](#)

Các file được tải xuống thành công, lúc này màn hình console sẽ thông báo tải xuống hoàn tất:



```
C:\WINDOWS\system32\cmd.exe
--> Got a request
video.mp4 downloading...
Completed!
Client ('127.0.0.1', 54489) connected!
--> Got a request
picture.jpg downloading...
Completed!
Client ('127.0.0.1', 54490) connected!
Client ('127.0.0.1', 54491) connected!
--> Got a request
text.txt downloading...
Completed!
--> Got a request
text.txt downloading...
Completed!
Client ('127.0.0.1', 54492) connected!
--> Got a request
music.mp3 downloading...
Client ('127.0.0.1', 54493) connected!
Completed!
--> Got a request
music.mp3 downloading...
Completed!
```

## IV. DANH MỤC TÀI LIỆU THAM KHẢO

1. Tài liệu Socket (Python) – Moodle FIT.HCMUS
2. HTML Tutorial: <https://www.w3schools.com/html/>
3. Transfer-Encoding: <https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Transfer-Encoding>
4. Thread trong Python:  
<https://viblo.asia/p/da-luong-trong-python-multithreading-WAyK8MO6ZxX>  
<https://realpython.com/intro-to-python-threading/>