

**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**  
**TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN**  
**NHÓM NGÀNH MÁY TÍNH VÀ CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO ĐỒ ÁN**  
**LẬP TRÌNH SOCKET**

Bộ môn : Mạng máy tính  
Giảng viên : Lê Ngọc Sơn  
Lớp : 18CTT5  
Sinh viên :

Ngô Trung Tín  
Nguyễn Phan Công Vương

MSSV: 18120596  
MSSV: 18120654

**Thành phố Hồ Chí Minh, tháng 07 năm 2020**

## MỤC LỤC

<b>I.</b>	<b>Đề bài &amp; phân công công việc.....</b>	<b>1</b>
<b>1.</b>	<b>Đề bài .....</b>	<b>1</b>
<b>2.</b>	<b>Phân công công việc .....</b>	<b>1</b>
<b>II.</b>	<b>Nội dung thực hiện .....</b>	<b>1</b>
<b>1.</b>	<b>Thư viện chính .....</b>	<b>1</b>
<b>2.</b>	<b>Các hàm thực thi .....</b>	<b>2</b>
<b>3.</b>	<b>Thực thi chương trình.....</b>	<b>3</b>
<b>III.</b>	<b>Mức độ hoàn thành .....</b>	<b>7</b>
<b>IV.</b>	<b>Tài liệu tham khảo.....</b>	<b>7</b>

## I. Đề bài & phân công công việc

### 1. Đề bài

Sử dụng lập trình Socket để viết một ứng dụng Web Server trả về nội dung các trang web:

- *index.html*: Trang yêu cầu người dùng nhập username và password để đăng nhập
- *info.html*: Khi người dùng nhập đúng thông tin đăng nhập tại trang đăng nhập (admin/admin) thì redirect đến trang *info.html*, trang này gồm hình ảnh và thông tin của các thành viên trong nhóm.
- *404.html*: Nếu người dùng nhập sai thông tin đăng nhập thì trả về trang *404.html* thông báo lỗi.

### 2. Phân công công việc

Họ tên	MSSV	Công việc
Ngô Trung Tín	18120596	<ul style="list-style-type: none"><li>- Viết ứng dụng Web Server trả về nội dung trang <i>index.html</i>.</li><li>- Thiết kế các trang <i>index.html</i>, <i>info.html</i>, <i>404.html</i>.</li><li>- Viết báo cáo.</li></ul>
Nguyễn Phan Công Vương	18120654	<ul style="list-style-type: none"><li>- Xử lý username/password để chuyển hướng đến trang <i>info.html</i> hoặc <i>404.html</i>.</li><li>- Viết báo cáo</li></ul>

## II. Nội dung thực hiện

### 1. Thư viện chính

- Sử dụng thư viện **winsock2.h** và **WS2tcpip.h**, khai báo thêm thư viện **ws2\_2.lib**

```
1  #define _WINSOCK_DEPRECATED_NO_WARNINGS
2  #define _CRT_SECURE_NO_WARNINGS
3
4  #include <winsock2.h>
5  #include <string>
6  #include <vector>
7  #include <sstream>
8  #include <fstream>
9  #include <io.h>
10 #include <iostream>
11 #include <WS2tcpip.h>
12
13 #pragma comment (lib, "ws2_32.lib")
14
15 using namespace std;
```

## 2. Các hàm thực thi

- Tạo class Server.

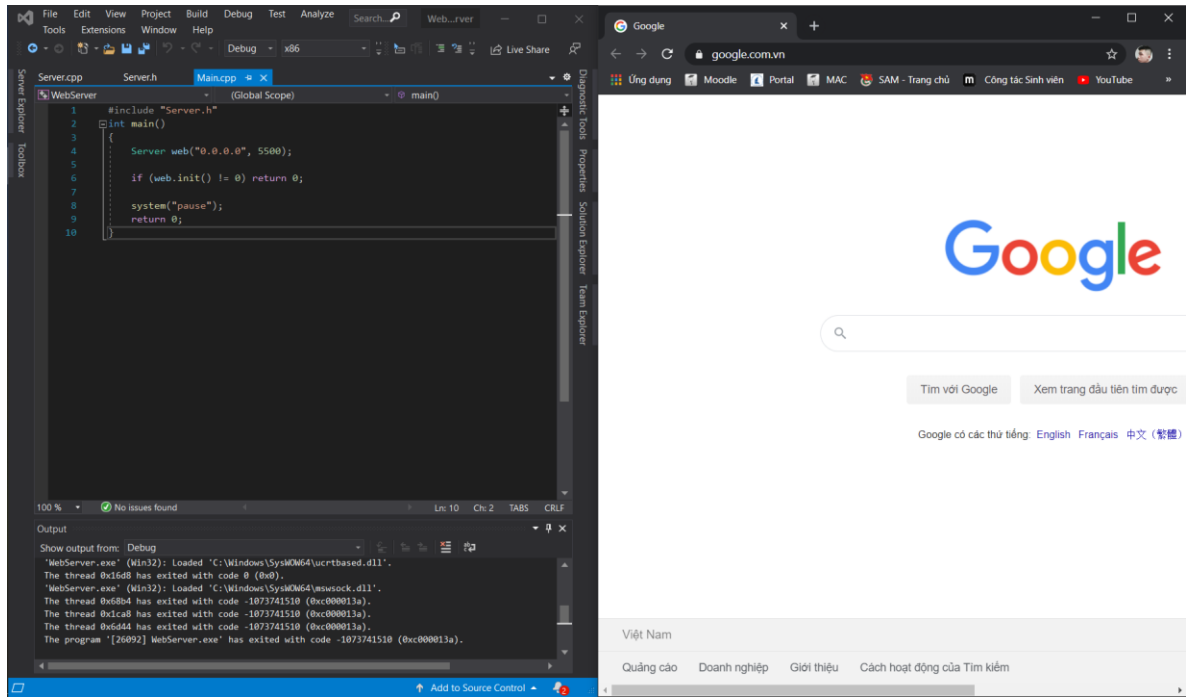
```
17 class Server
18 {
19     private:
20         const char* IP;        // IP address
21         int port;              //Port for the webserver
22         SOCKET sock;
23         char* CHECK;           //IF USERNAME&PSW INCORRECT, RETURN NULL
24     public:
25         //Constructor but just IP and Port
26         Server(const char* ip, int p);
27
28         //Initialize the server
29         int init();
30
31         //Handle the connection to client
32         void HandleConnection(SOCKET s, const char* msg, int len);
33     };
```

- Hàm tạo **Server**(*const char\* ip, int p*): Khởi tạo các giá trị IP và port cho server
- Hàm khởi tạo **int init()**: Khởi tạo các chức năng cần thiết cho server
  - Khởi tạo winsock
  - Khởi tạo socket
  - Bind địa chỉ IP và Port đến socket
  - Bắt đầu lắng nghe kết nối tại port vừa bind
  - Chấp nhận kết nối
  - Khi kết nối vẫn đang bật, nhận thông tin từ client, kiểm tra cò CHECK username và password. Nếu thông tin nhận lỗi thì đóng kết nối. Nếu nhận thành công thì thực hiện xử lý giao tiếp bằng hàm HandleConnection.
- Hàm xử lý **void HandleConnection**(*SOCKET s, const char\* msg, int len*): Xử lý các giao tiếp giữa Client và Server khi kết nối.
  - Tạo một vector kiểu string tên là **analy** để lưu thông tin nhận từ client.
  - Client sẽ gửi lên server thông tin yêu cầu kết nối có dạng: *GET /index.html HTTP/1.1* .....
  - Ta xét vector **analy**, nếu chiều dài vector  $\geq 3$  và phần tử đầu tiên của vector là “GET”, phần tử thứ hai là “/” hoặc “/index.html”, ta gán cho biến kiểu string **htmlFile** là “index.html”. Nếu phần tử đầu tiên là “POST” và cò CHECK khác rỗng (Username và password chính xác), ta gán cho biến **htmlFile** là “info.html”. Nếu không thỏa một trong hai điều kiện trên, ta gán biến **htmlFile** là “404.html”
  - Ta bắt đầu mở các file html trong thư mục *.\www\htmlFile* lưu trong project. Nếu mở thành công, ta lưu toàn bộ nội dung của **htmlFile** vào biến kiểu string tên là **str**.
  - Sau đó ta bắt đầu gửi response về client theo cú pháp: *HTTP/1.1 200 OK\r\n...* Dùng hàm send().

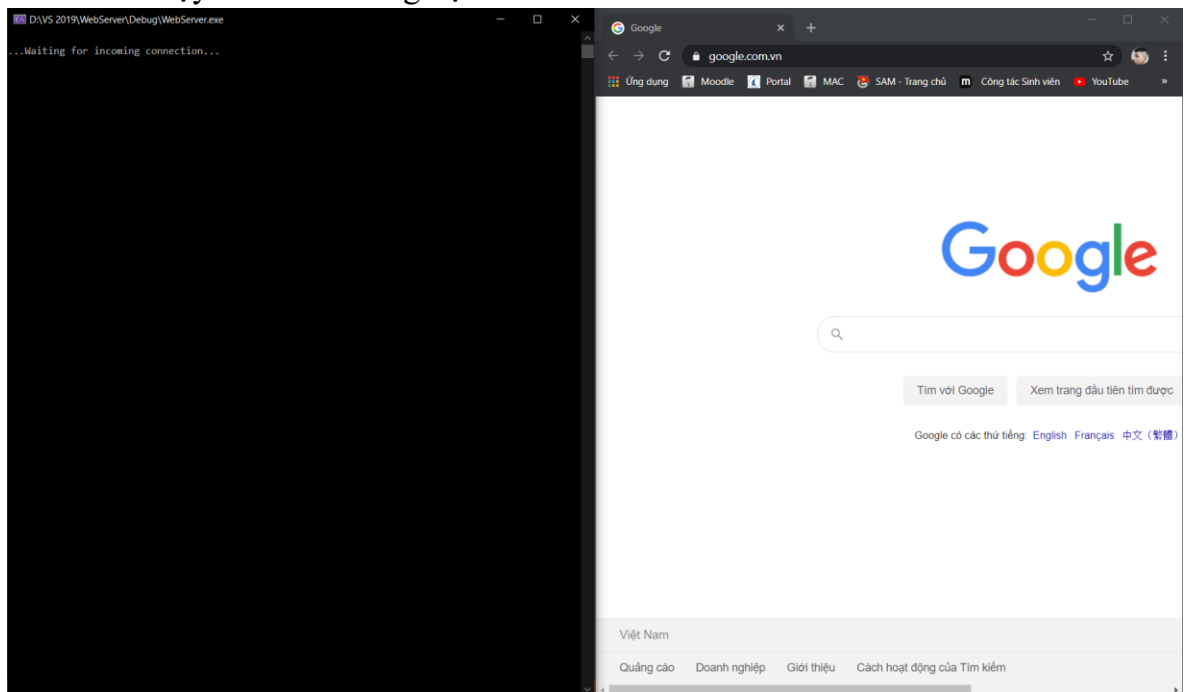
### 3. Thực thi chương trình

Ta sử dụng port **5500** và client kết nối tới server thông qua url **localhost:5500**

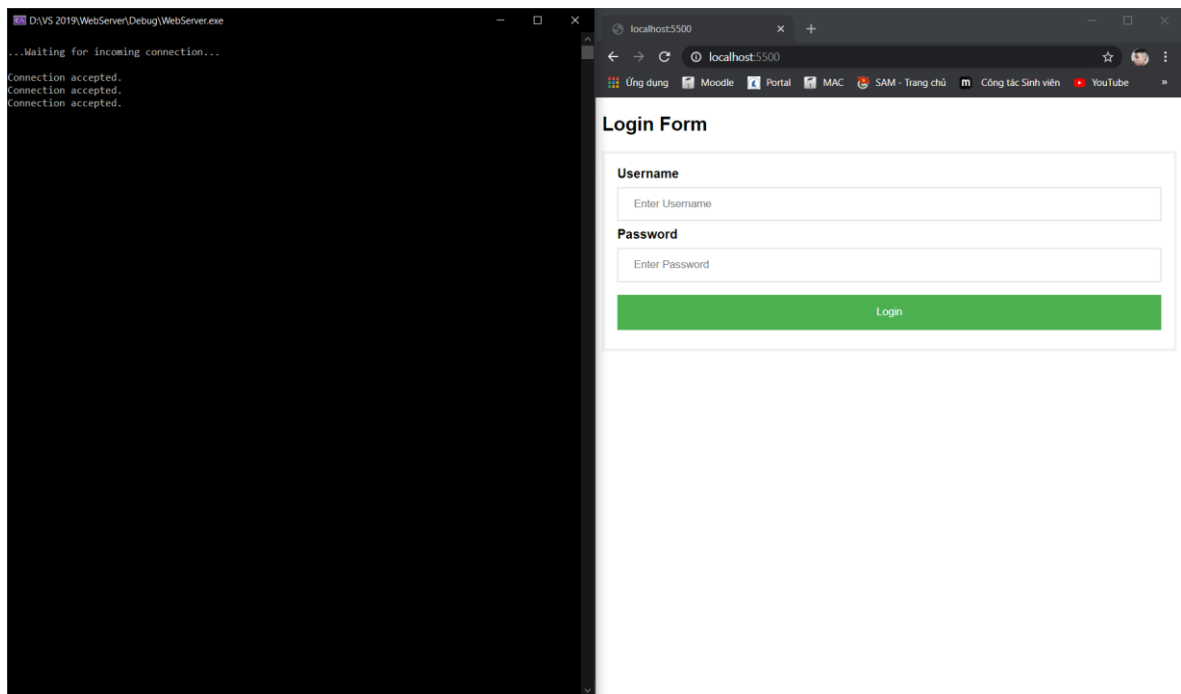
- Trước khi thực thi:



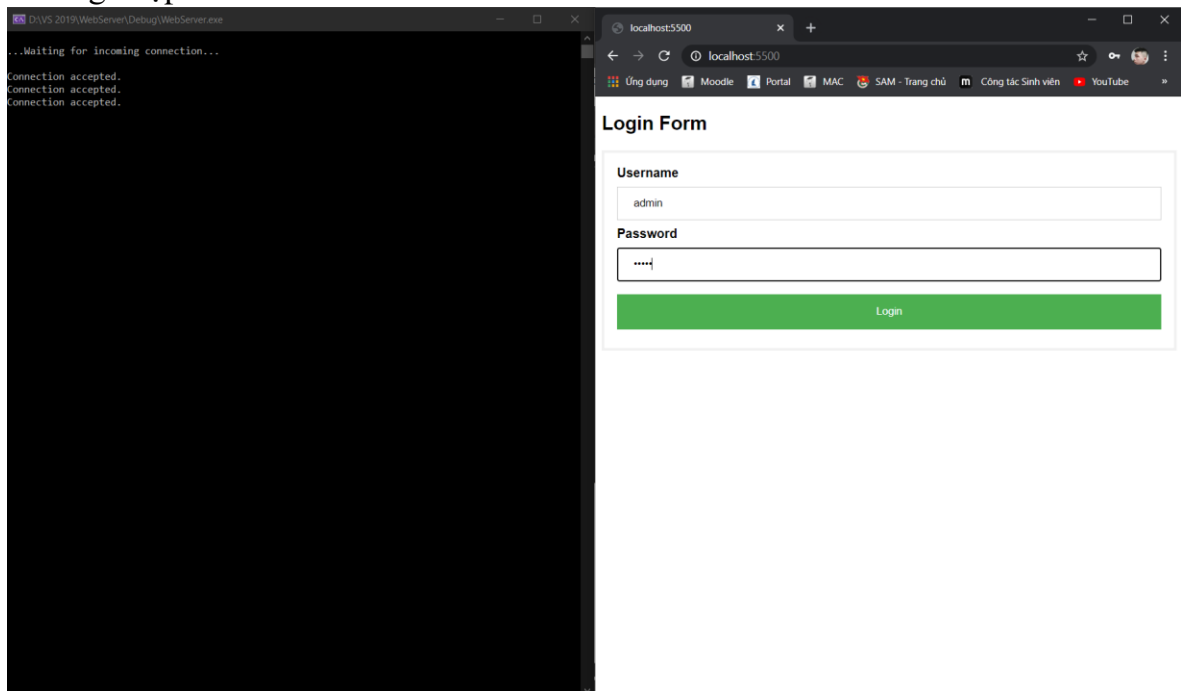
- Sau khi chạy và server đang đợi kết nối từ client:



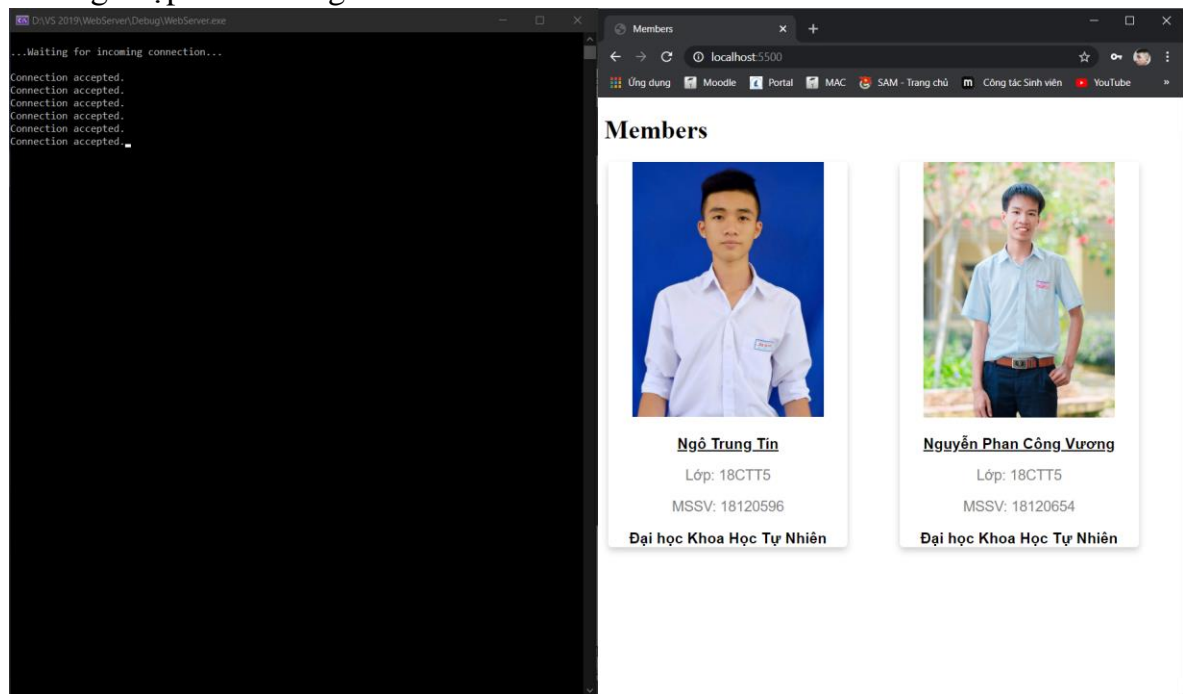
- Client và server đã kết nối:



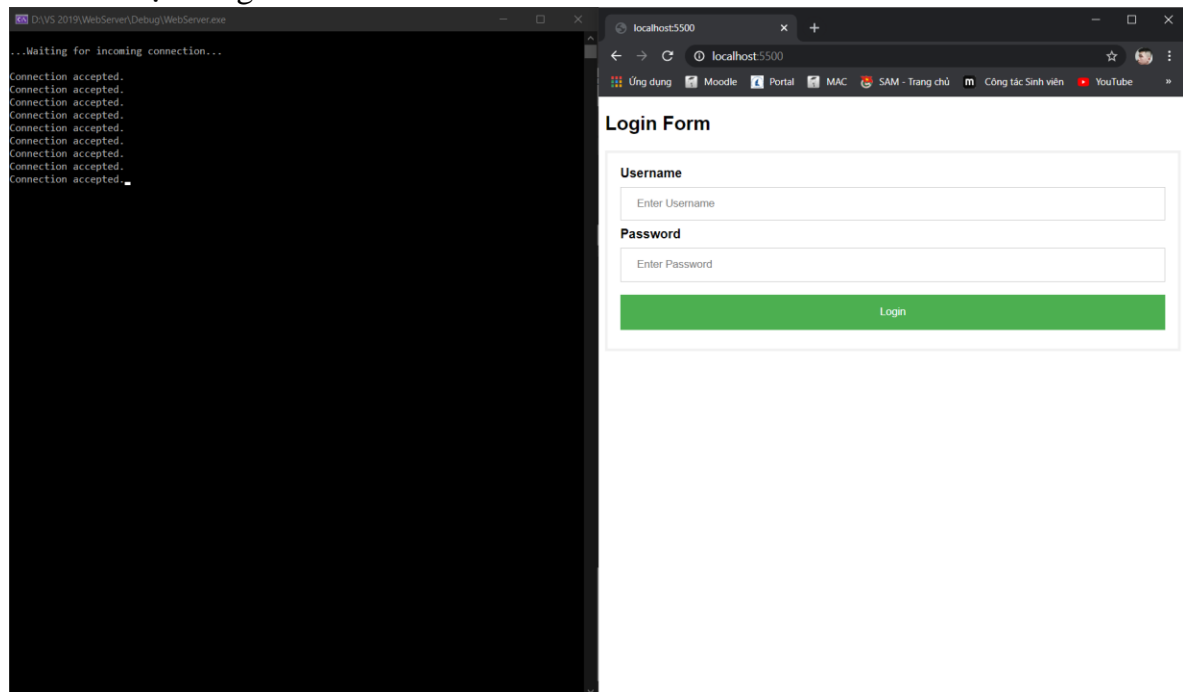
- Đăng nhập:



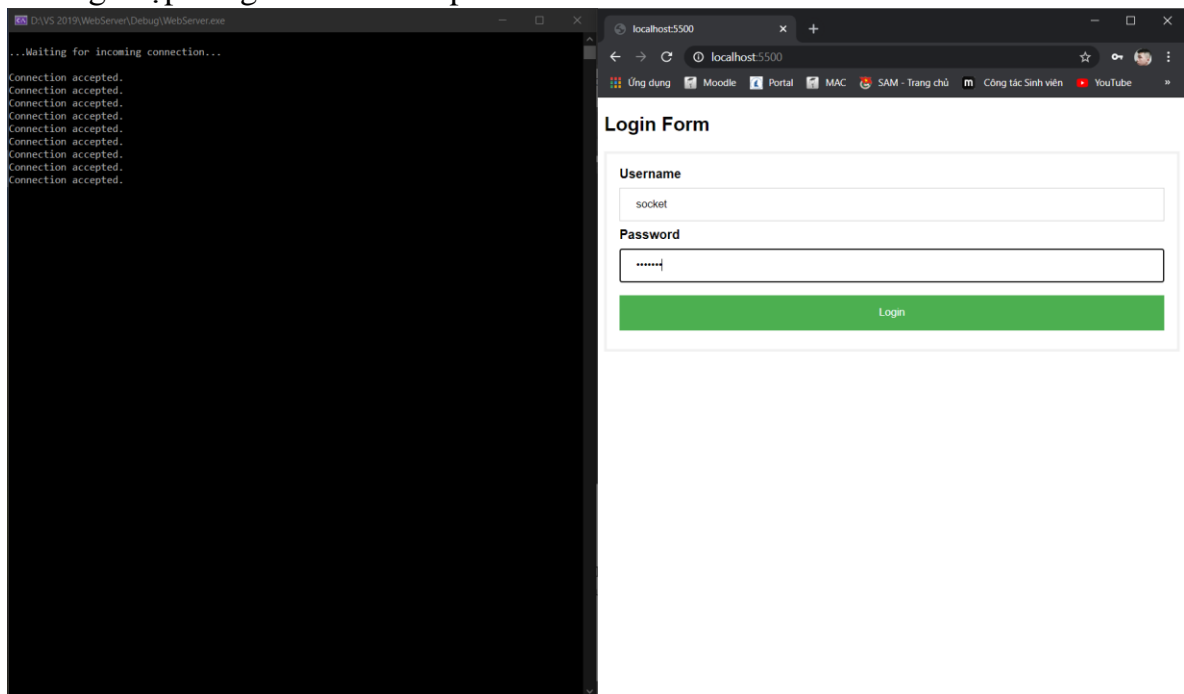
- Đăng nhập thành công:



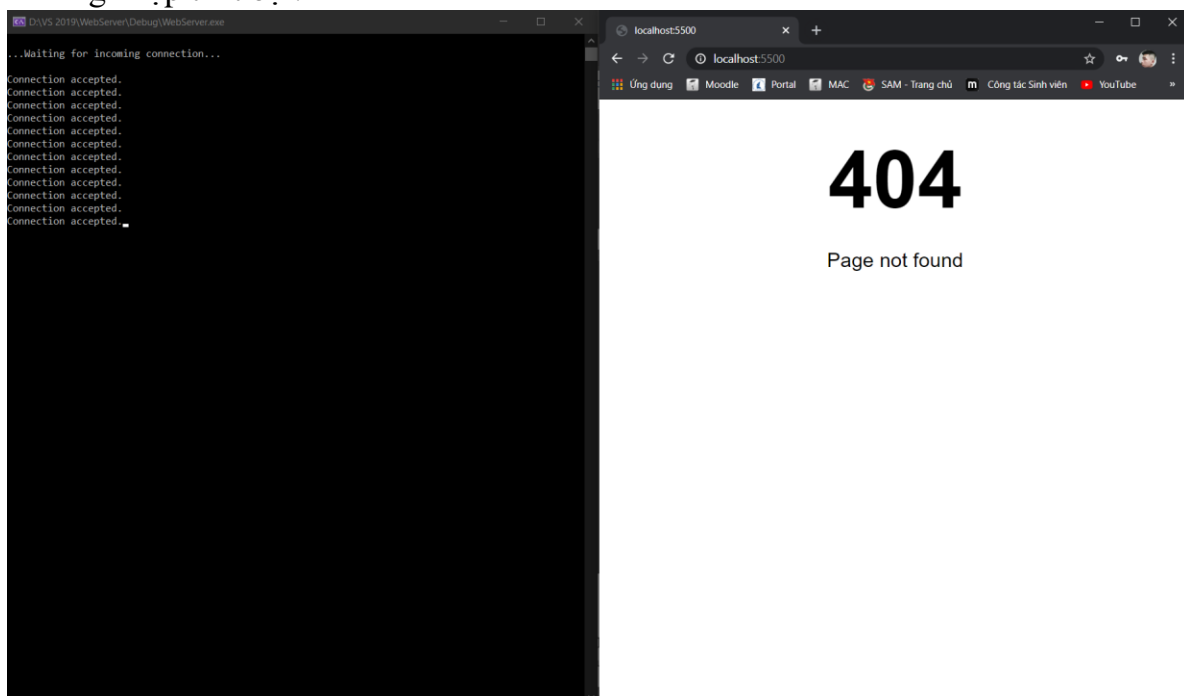
- Reload lại trang web:



- Đăng nhập bằng username và password sai:



- Đăng nhập thất bại:



- Video thực thi chương trình:

[https://drive.google.com/file/d/1\\_IKpN4vCQHDwtXq3-9Q\\_IBDnqdlZWzgr/view?usp=sharing](https://drive.google.com/file/d/1_IKpN4vCQHDwtXq3-9Q_IBDnqdlZWzgr/view?usp=sharing)



### **III. Mức độ hoàn thành**

Hoàn thành 100% các yêu cầu đặt ra của đề bài.

### **IV. Tài liệu tham khảo**

- Slide bài giảng Mạng Máy Tính.
- <https://developer.mozilla.org/en-US/docs/Web/HTTP/Redirects>
- <https://docs.microsoft.com/en-us/windows/win32/api/winsock2/>
- [https://google.github.io/google-api-cpp-client/latest/guide/making\\_http\\_requests.html](https://google.github.io/google-api-cpp-client/latest/guide/making_http_requests.html)