Socket Programming and Web Development

Project No One

Friday, May 10, 2024

Group Members
leelyan karajah 1201191
Dareen muhanna 1202963
Section :4

Instructor : Dr. Ibrahim Nemer

Abstract

This project will teach us a variety of skills. We will track packets by using the "Command Prompt" program to run various commands, and we will use Wireshark to examine various DNS messages.
Then, since they communicate with one another over TCP connections, we will utilize socket programming to create a basic client-server system.
Ultimately, we will use socket programming to build a web server that can process many files and respond to various HTTP requests from the client. In addition to building an HTML and CSS web page with the details of the group members on it.

Contents

Part 1:

1. In your own words, what are ping, tracert, nslookup, and telnet (write one sentence for each one).

- **Ping:** Internet Package Grouper. It's the most popular utility tool for troubleshooting. Help to test host/server reachability, internet connection, network interface card, and DNS difficulties.
- **Traceroute:** It is used to determine the exact path that data takes on its way to the destination by sending Internet Control Message Protocol (ICMP) echo packets to it.
- **NSLOOKUP:** It is a utility that finds the IP address or DNS record of a given host name, and to find the attached domain for an IP address.
- **Telnet:** a network protocol that enables two-way collaborative and text-based communication between two machines, as well as the ability to connect to remote computers over a TCP/IP network.

---

2. Make sure that your computer is connected to the internet and then run the following commands:

A. Ping a device in the same network, e.g. from a laptop to a smartphone

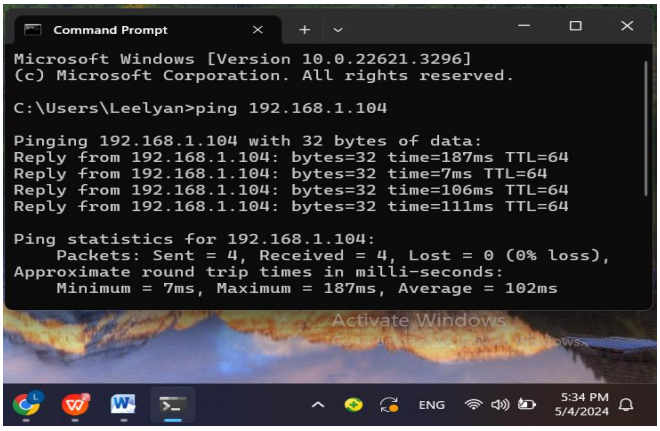


Figure 1: ping a device in the same network

We used cmd to test the ping command by sending four data packets to the targeted IP address (we used the IP of a cell phone) the output of this command contained the bytes of the data packets we sent which it was (32 bytes). Each one of the packets we sent had three parameters to describe it:

1- Size of each Packet: measured by bytes.
2- Time needed for each packet to send and received measured by milliseconds.
3- Time To Live (TTL): number of hops the packet is permitted to travel before being discarded.

In the above screenshot you can see that all the four packets we sent was successfully sent and received, the average round-trip time was 102 milliseconds

---

B. ping www.stanford.edu

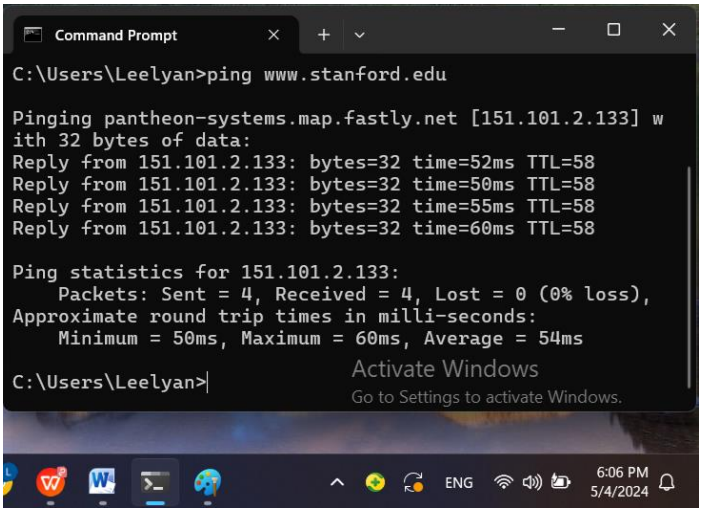


Figure 2: www.stanford.edu

We used cmd to preform ping www.stanford.edu , we saw the round-trip time (in milliseconds) for each packet, the IP address of "pantheon systems.map.fastly.net" is 151.101.2.133. It also shows the replay of each message

containing the size of each packet in bytes, the time from sending to receiving this packet and TTL which is the number of nods each packet has to pass through.

C. From the ping results, do you think the response you got is from USA? Explain your answer briefly.

The ping results alone, which measure Round Trip Time and network latency, do not provide information about the geographical location of the server associated with "www.stanford.edu" Determining the server's location requires additional methods such as geolocation services or querying the IP address using WHOIS services. By extracting the IP address (151.101.2.133) from the CMD prompt window and checking its approximate location on websites like iplocation.net or ipinfo.io, the results suggest a potential association with the USA, specifically Washington, or Canada, more precisely Quebec. However, it's crucial to emphasize that these findings offer a possible location, and the exact source cannot be definitively verified solely based on the ping results.

D. tracert www.stanford.edu

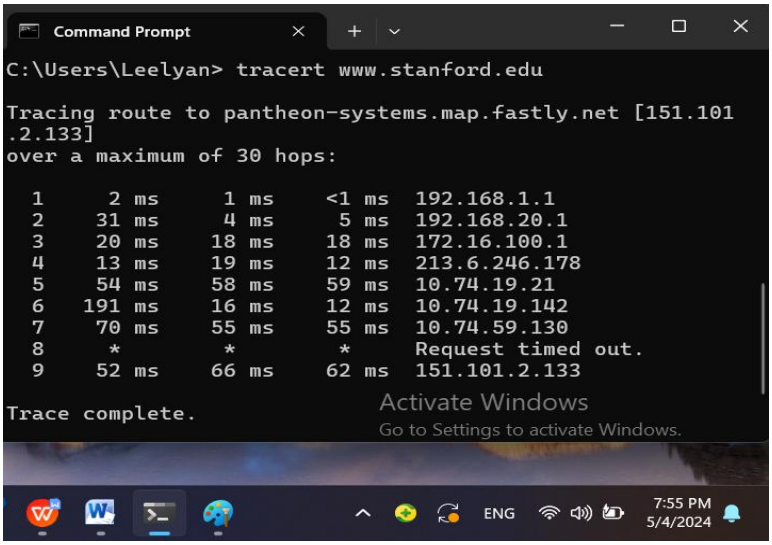


Figure 3 : www.stanford.edu

We used Cmd to preform tracert, the first line shows the target IP address or domain being traced and its corresponding IP address. The Column Headers:

- **Hop:** The number of the hop in the route.
- **Hostname (or IP address):** The IP address or hostname of the router at that hop.
- **Time (ms):** Round-trip time in milliseconds for the packet to reach that hop and back.

Rows represents a hop along the route. The first hop is typically our local router or gateway. Subsequent hops are intermediate routers or switches that the packet passes through. Times in milliseconds represent the round-trip time for a packet to reach that hop and return. The three times shown are the result of sending three separate packets to that hop.

For the "Request timed out" in a traceroute result, it means that the ICMP (Internet Control Message Protocol) echo request sent to the specific hop did not receive a response within the timeout period.

E. nslookup www.stanford.edu

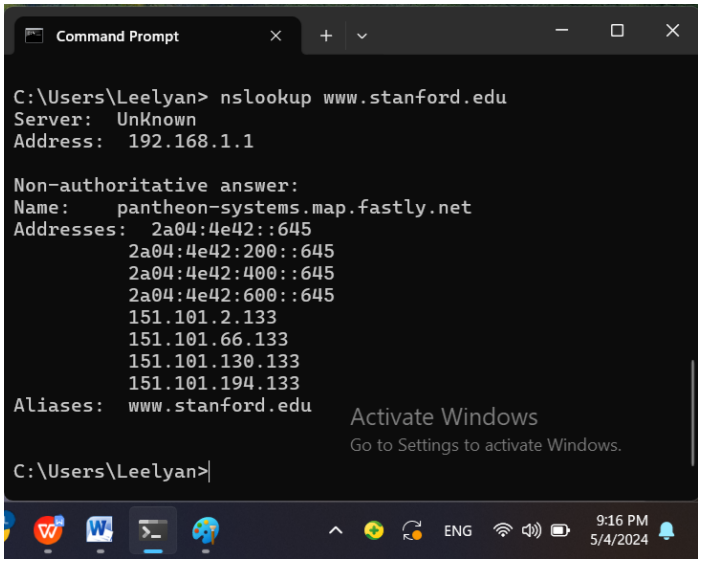


Figure 4: nslookup www.standford.edu

We used cmd to preform nslookup command, the output was shown as follow:

- **Server:** The DNS server that provided the resolution. In this example, it's "Unknown" with an IP address of "192.168.1.1,"

- **Non-authoritative answer:** This indicates that the information was obtained from a DNS server that is not the ultimate authority for the domain.
- **Name:** The resolved domain name. In this case, “www.stanford.edu " resolves to "pantheon-systems.map.fastly.net".
- **Addresses:** The IP addresses associated with the resolved domain.
- **Aliases:** Additional domain names associated with the resolved IP addresses.

---

3. use wireshark to capture some DNS messages.

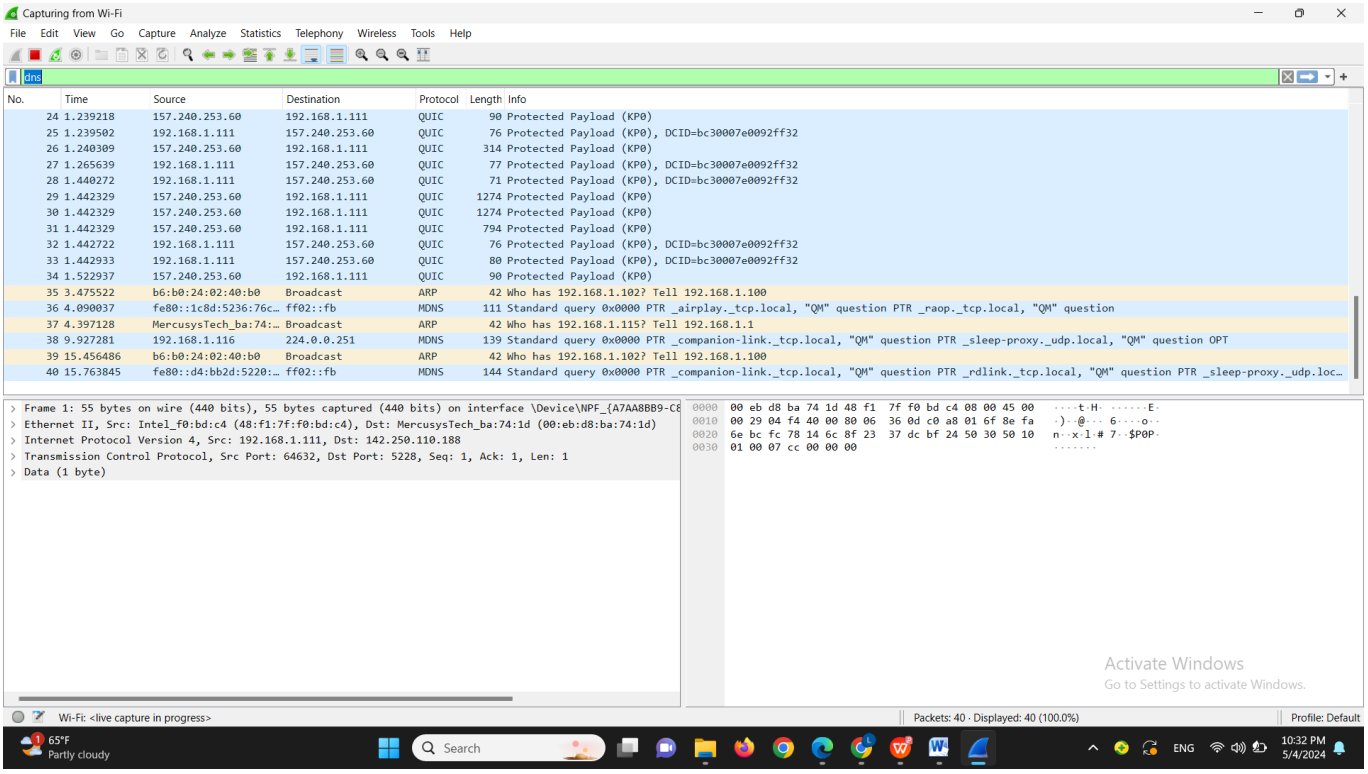


Figure 5: capture some DNS messages

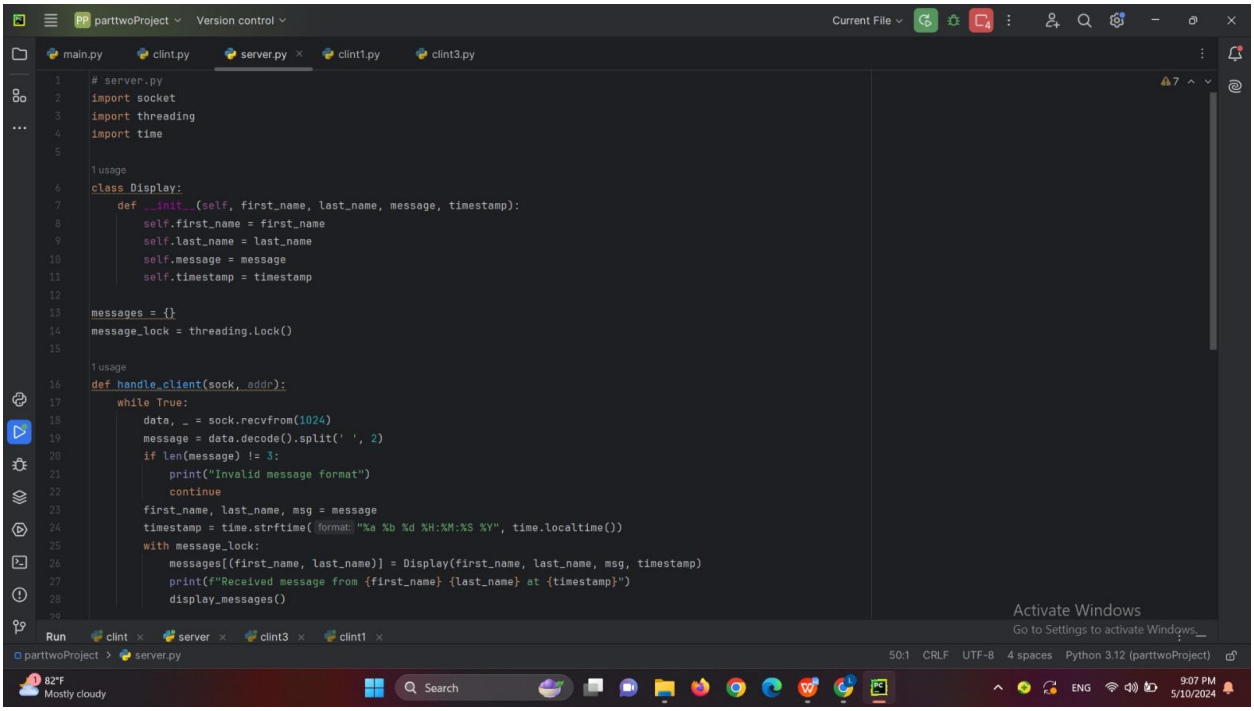
When capturing DNS (Domain Name System) messages in Wireshark, we can see DNS request and response packets, DNS record types, DNS Queries for local domains, DNS flags, etc.

DNS protocol translates human readable domain names into IP addresses.

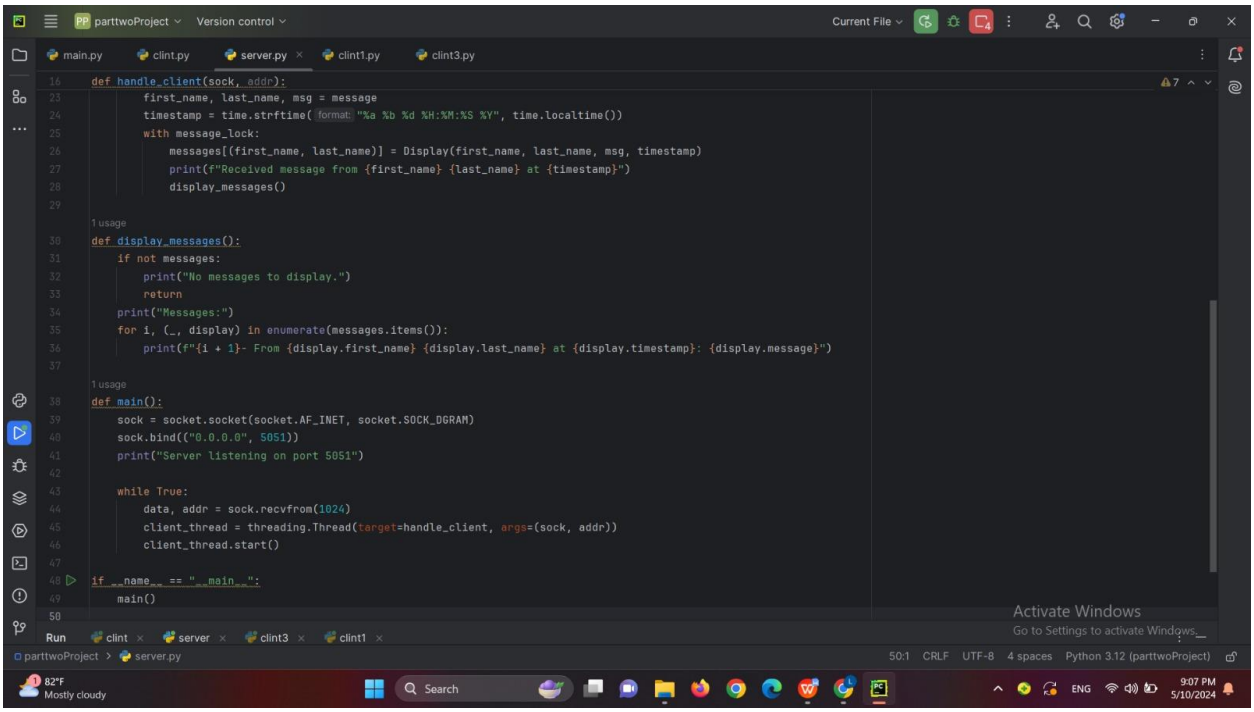
Part 2:

Server Code

Figure 6:server codes



```
# server.py
import socket
import threading
import time

1 usage
class Display:
    def __init__(self, first_name, last_name, message, timestamp):
        self.first_name = first_name
        self.last_name = last_name
        self.message = message
        self.timestamp = timestamp

messages = {}
message_lock = threading.Lock()

1 usage
def handle_client(sock, addr):
    while True:
        data, _ = sock.recvfrom(1024)
        message = data.decode().split(' ', 2)
        if len(message) != 3:
            print("Invalid message format")
            continue
        first_name, last_name, msg = message
        timestamp = time.strftime( format: "%a %b %d %H:%M:%S %Y", time.localtime())
        with message_lock:
            messages[(first_name, last_name)] = Display(first_name, last_name, msg, timestamp)
            print(f"Received message from {first_name} {last_name} at {timestamp}")
            display_messages()
```



```
def handle_client(sock, addr):
        first_name, last_name, msg = message
        timestamp = time.strftime( format: "%a %b %d %H:%M:%S %Y", time.localtime())
        with message_lock:
            messages[(first_name, last_name)] = Display(first_name, last_name, msg, timestamp)
            print(f"Received message from {first_name} {last_name} at {timestamp}")
            display_messages()


1 usage
def display_messages():
    if not messages:
        print("No messages to display.")
        return
    print("Messages:")
    for i, (_, display) in enumerate(messages.items()):
        print(f"{i + 1}- From {display.first_name} {display.last_name} at {display.timestamp}: {display.message}")


1 usage
def main():
    sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
    sock.bind(("0.0.0.0", 5051))
    print("Server listening on port 5051")

    while True:
        data, addr = sock.recvfrom(1024)
        client_thread = threading.Thread(target=handle_client, args=(sock, addr))
        client_thread.start()

if __name__ == "__main__":
    main()
```

Explanation of server code

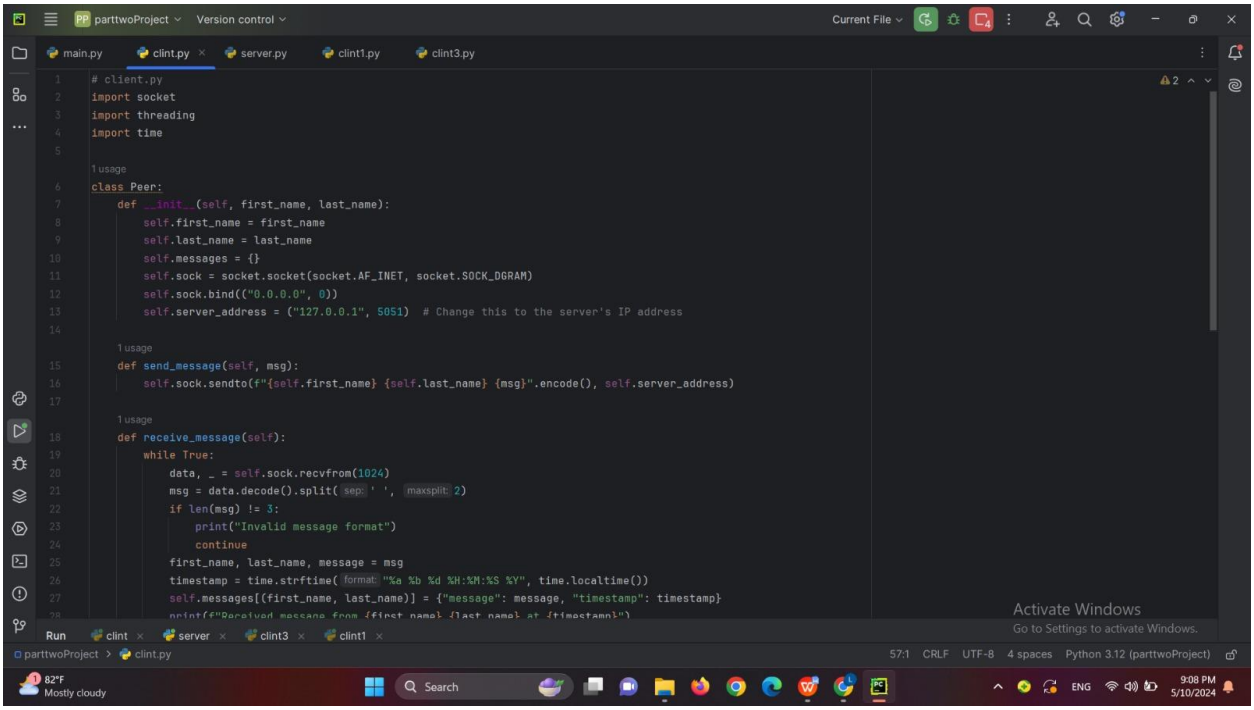
This simple Python script functions, as a UDP server, monitoring and showing messages. It can handle clients simultaneously with each client being handled through threads.

Using Sockets for Networking: The script creates a UDP server to receive incoming data. This makes it lighter, but less dependable than TCP.
Threading for concurrency:
The script uses threads to allow multiple clients to transmit messages concurrently without causing problems. Each incoming message is handled by its own thread, allowing the server to process numerous messages concurrently.
Data Storage:
The server saves receiving messages in a messages dictionary with a key that includes the sender's first and last name. This storage allows the server to keep track of every message and display them
Threading occurs when many threads simultaneously ask the messages dictionary.The lock is used to ensure that only one thread can edit a dictionary at a time. This prevents from data loss and keeps thread safety.

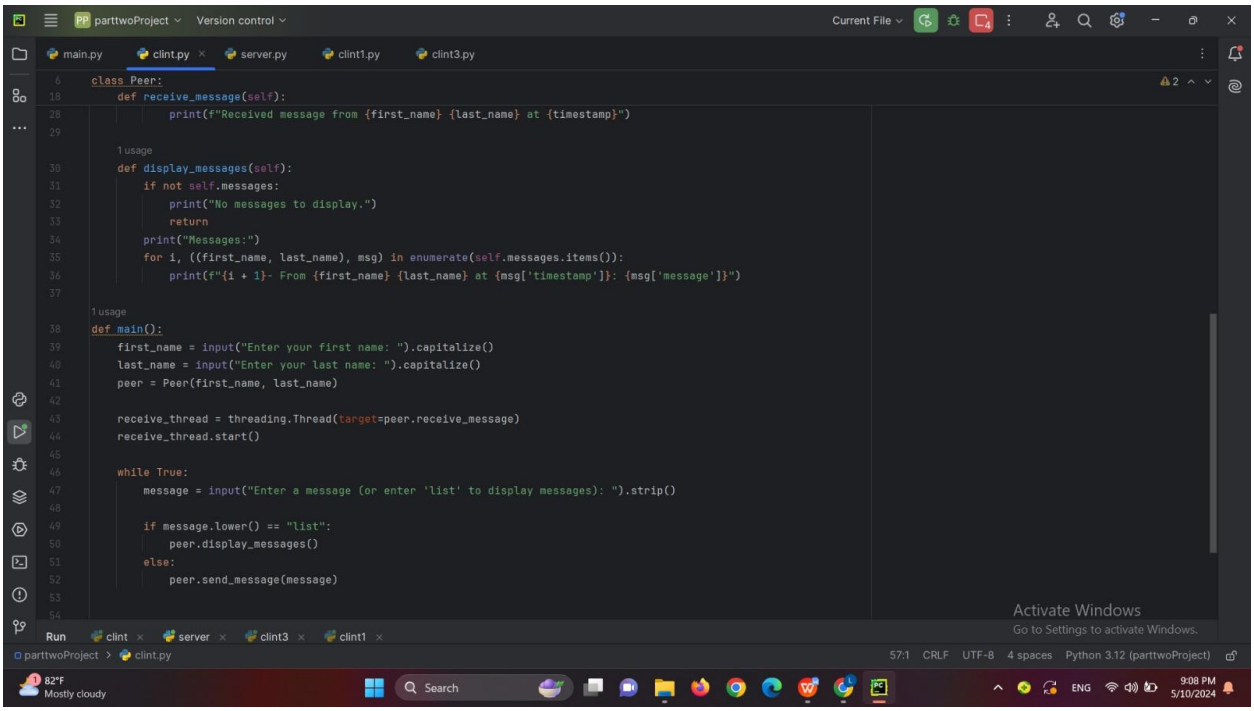
The handle_client function listens for messages decodes them. Gathers the senders first and last names, along, with the message content. It also creates a timestamp to show when the message was received.
After ensuring thread safety using a lock the function adds the message to the messages dictionary. Then calls display_messages() to show the updated message list.
Showing messages;
The display_messages function outputs all saved messages on the screen. If there are no messages it indicates that the message list is empty.

Client Code

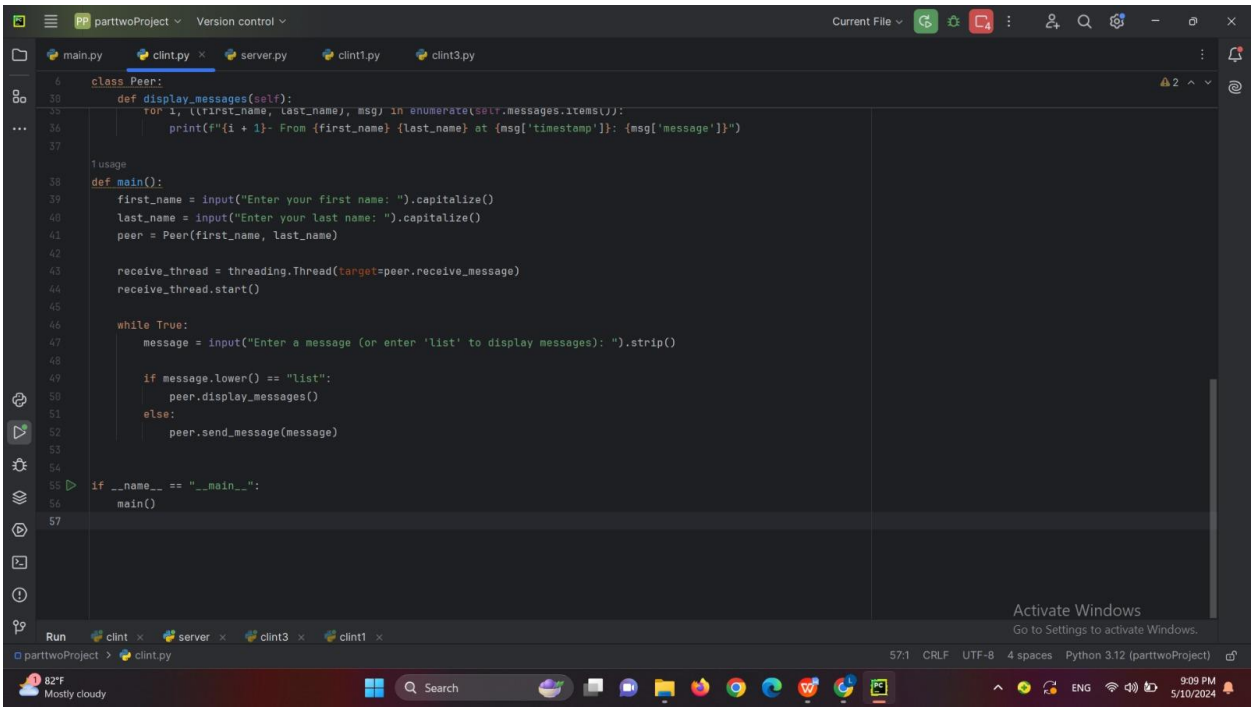
Figure 7:client code



```
# client.py
import socket
import threading
import time

1 usage
class Peer:
    def __init__(self, first_name, last_name):
        self.first_name = first_name
        self.last_name = last_name
        self.messages = {}
        self.sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
        self.sock.bind(("0.0.0.0", 0))
        self.server_address = ("127.0.0.1", 5051)  # Change this to the server's IP address

    1 usage
    def send_message(self, msg):
        self.sock.sendto(f"{self.first_name} {self.last_name} {msg}".encode(), self.server_address)

    1 usage
    def receive_message(self):
        while True:
            data, _ = self.sock.recvfrom(1024)
            msg = data.decode().split( sep: ' ', maxsplit: 2)
            if len(msg) != 3:
                print("Invalid message format")
                continue
            first_name, last_name, message = msg
            timestamp = time.strftime( format: "%a %b %d %H:%M:%S %Y", time.localtime())
            self.messages[(first_name, last_name)] = {"message": message, "timestamp": timestamp}
            print(f"Received message from {first_name} {last_name} at {timestamp}")
```



```
class Peer:
    def receive_message(self):
            print(f"Received message from {first_name} {last_name} at {timestamp}")

    1 usage
    def display_messages(self):
        if not self.messages:
            print("No messages to display.")
            return
        print("Messages:")
        for i, ((first_name, last_name), msg) in enumerate(self.messages.items()):
            print(f"{i + 1}- From {first_name} {last_name} at {msg['timestamp']}: {msg['message']}")

1 usage
def main():
    first_name = input("Enter your first name: ").capitalize()
    last_name = input("Enter your last name: ").capitalize()
    peer = Peer(first_name, last_name)

    receive_thread = threading.Thread(target=peer.receive_message)
    receive_thread.start()

    while True:
        message = input("Enter a message (or enter 'list' to display messages): ").strip()

        if message.lower() == "list":
            peer.display_messages()
        else:
            peer.send_message(message)
```



Explanation of client code

The Peer Class

Initializes with the peer's first and last name, plus an empty dictionary for storing messages.

Creates a UDP socket and assigns it a random port.

Defines the server address to which messages should be sent (in this scenario, localhost at port 5051 is assumed).

send_message(msg): Provides a message to the server. The message includes the sender's first and last name, and also the message content.

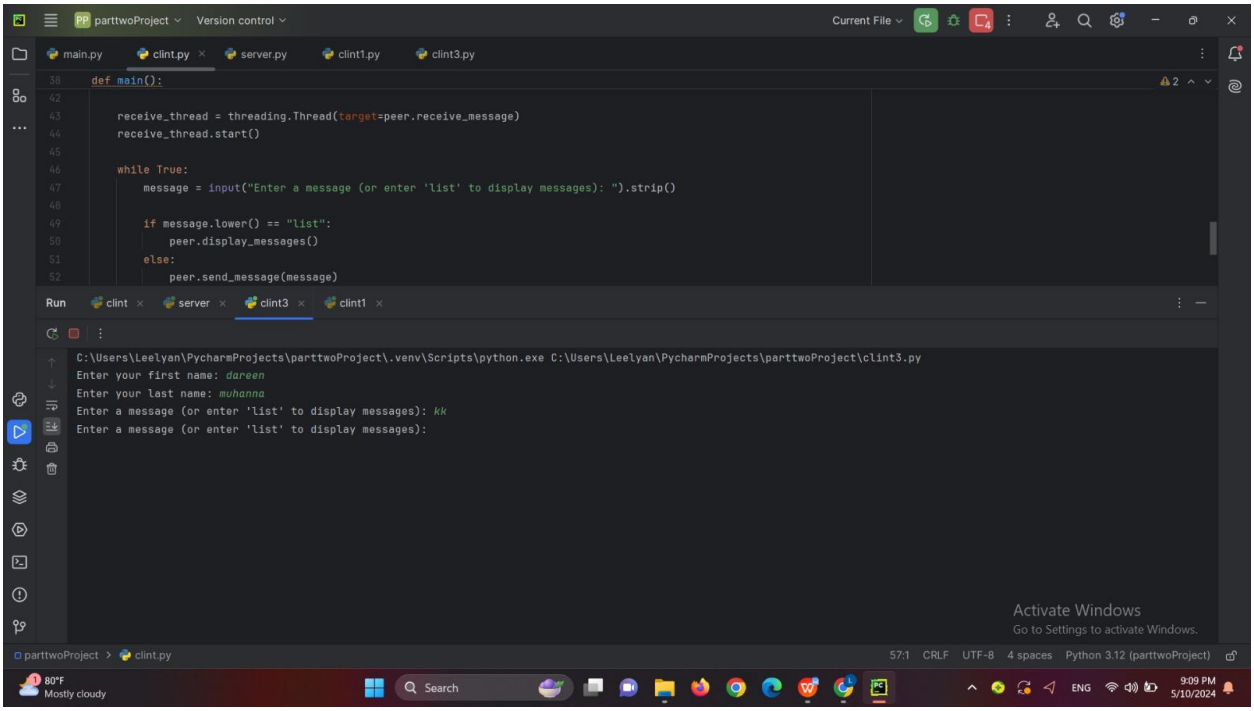
receive_message(): Responds regularly for incoming messages from the server. When a message is received, it takes the sender's name, message content, and timestamp before storing it in the peer's message dictionary.

display_messages() shows all messages sent by the peer.

In The main method :The script creates a Peer class object, listens for incoming messages, and enters a loop for sending and displaying received messages. If the user enters "list", the script displays the received messages.

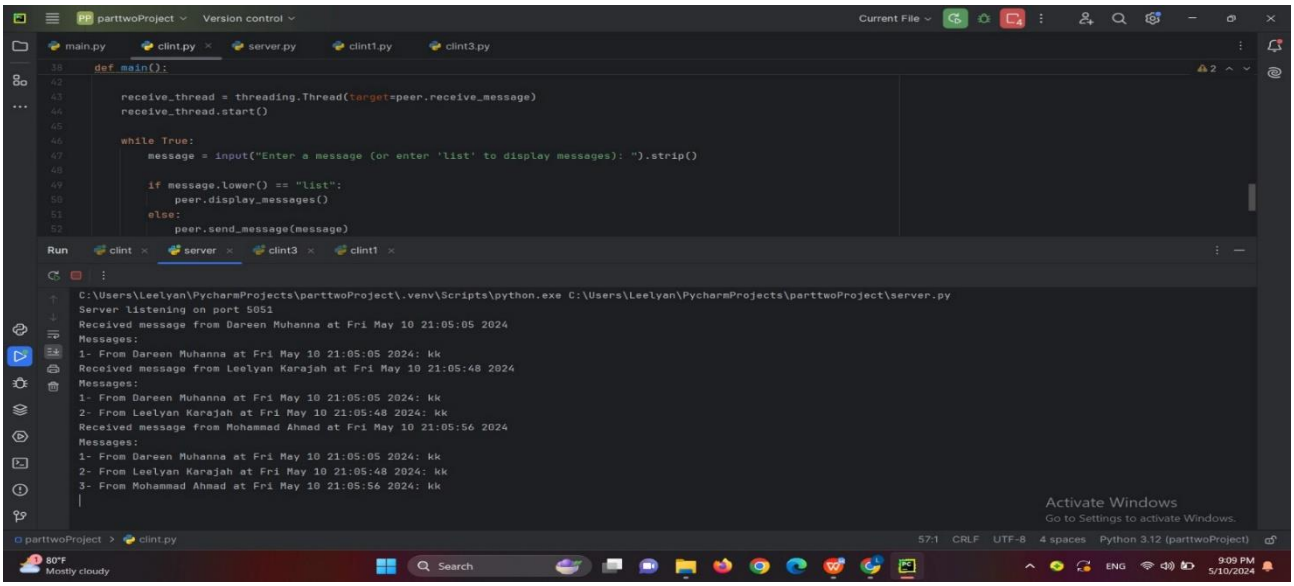
Client Input code

Figure 8:client input



Server Output code

Figure 9:server output code

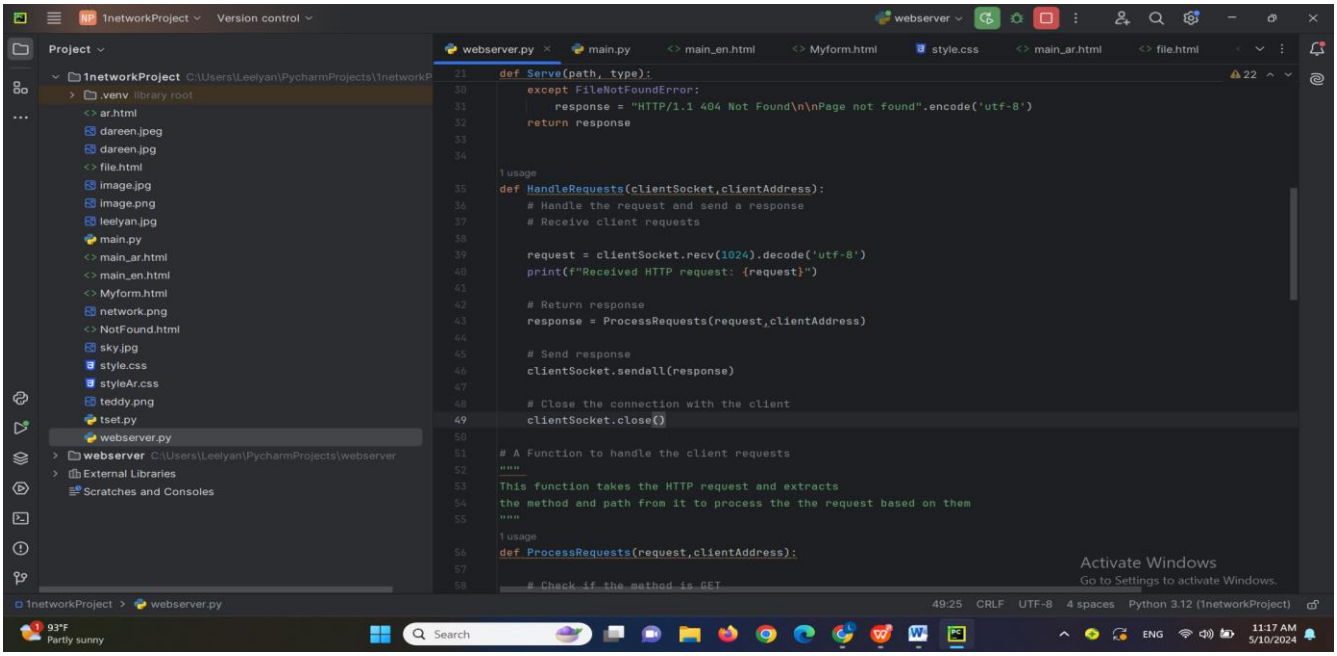


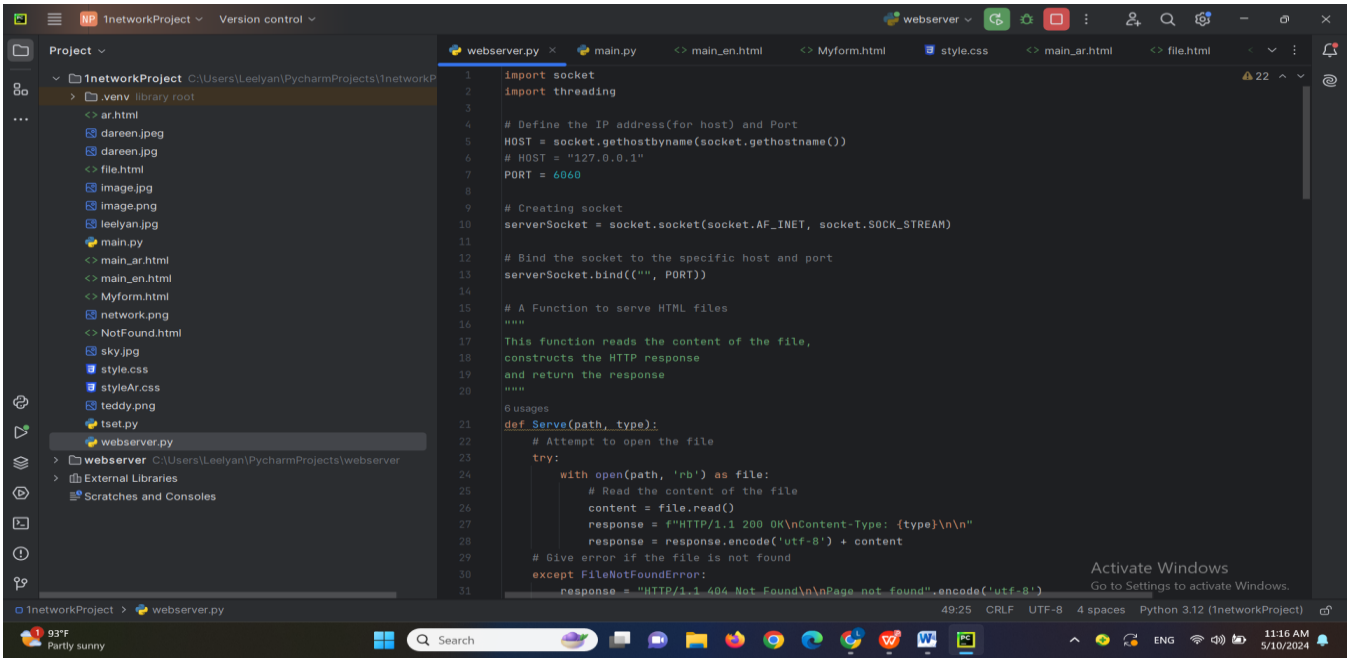
Part 3:

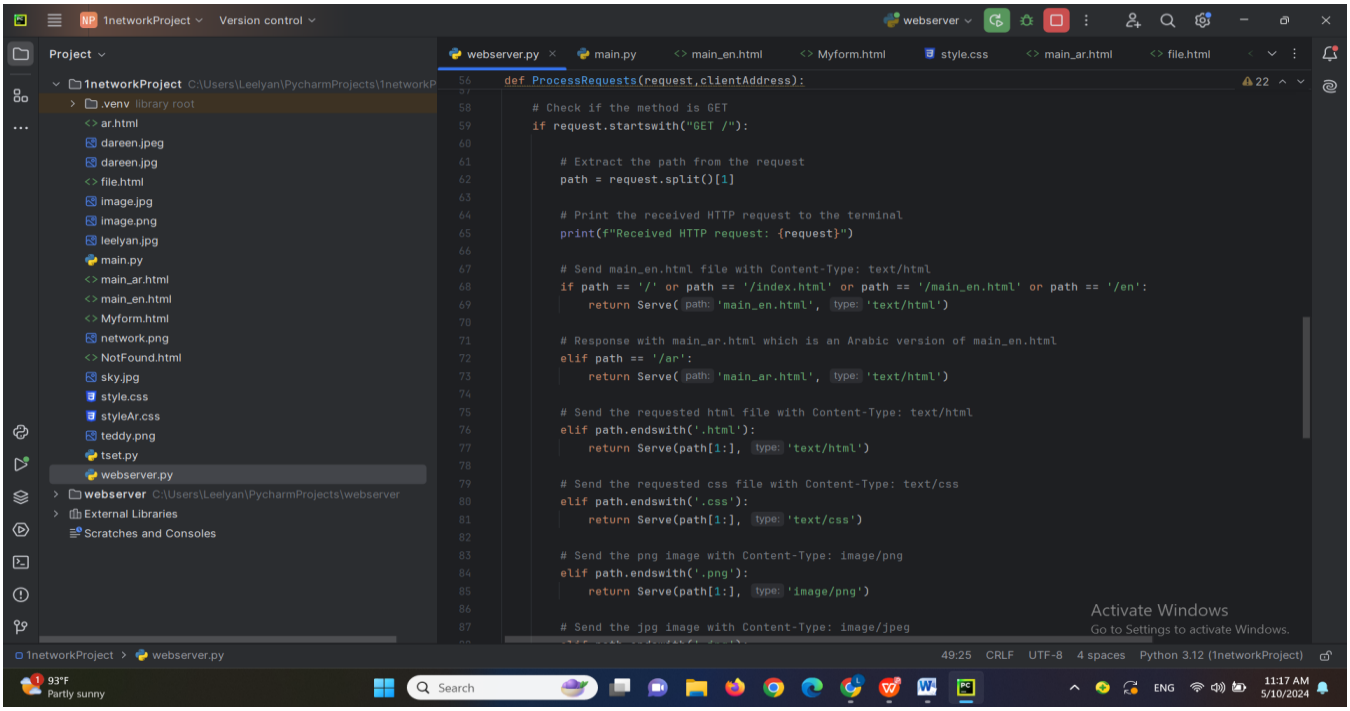
Using socket programming, implement a simple but a complete web server in go, python, java or C that is listening on port 6060. Make the code as general as possible

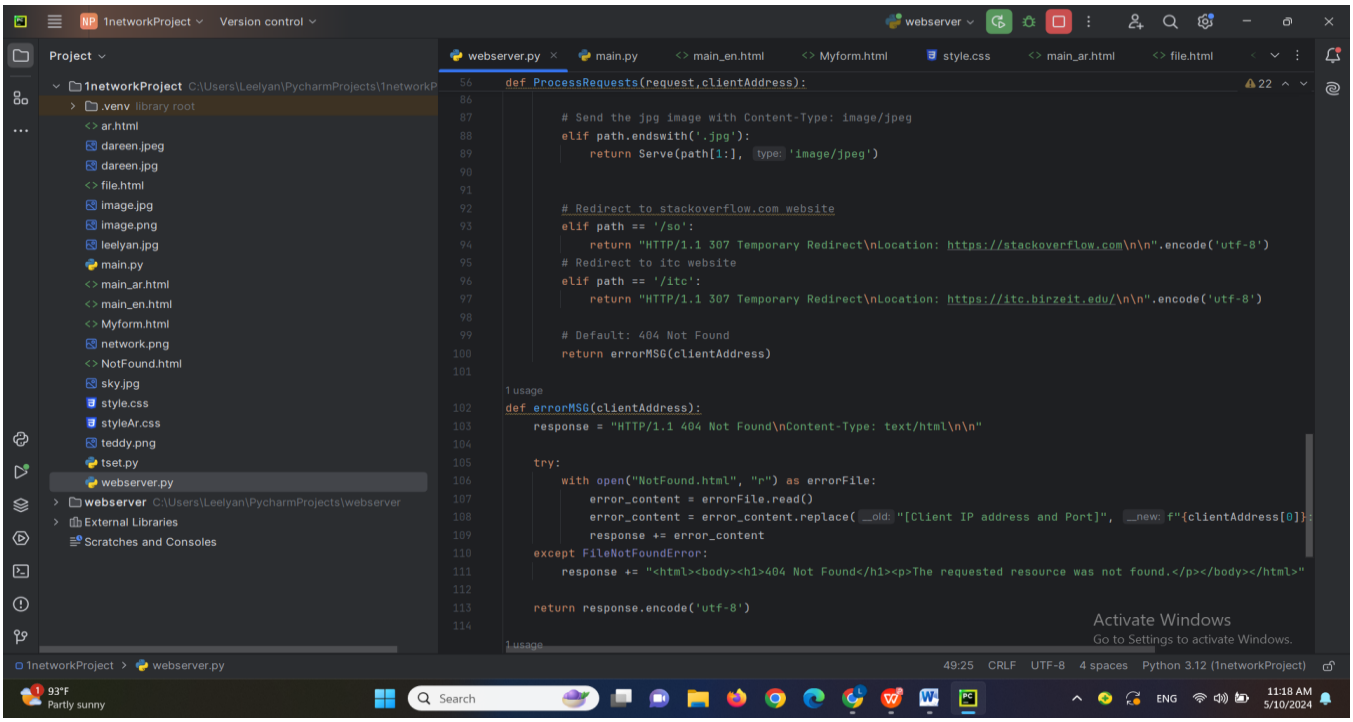
Web Server

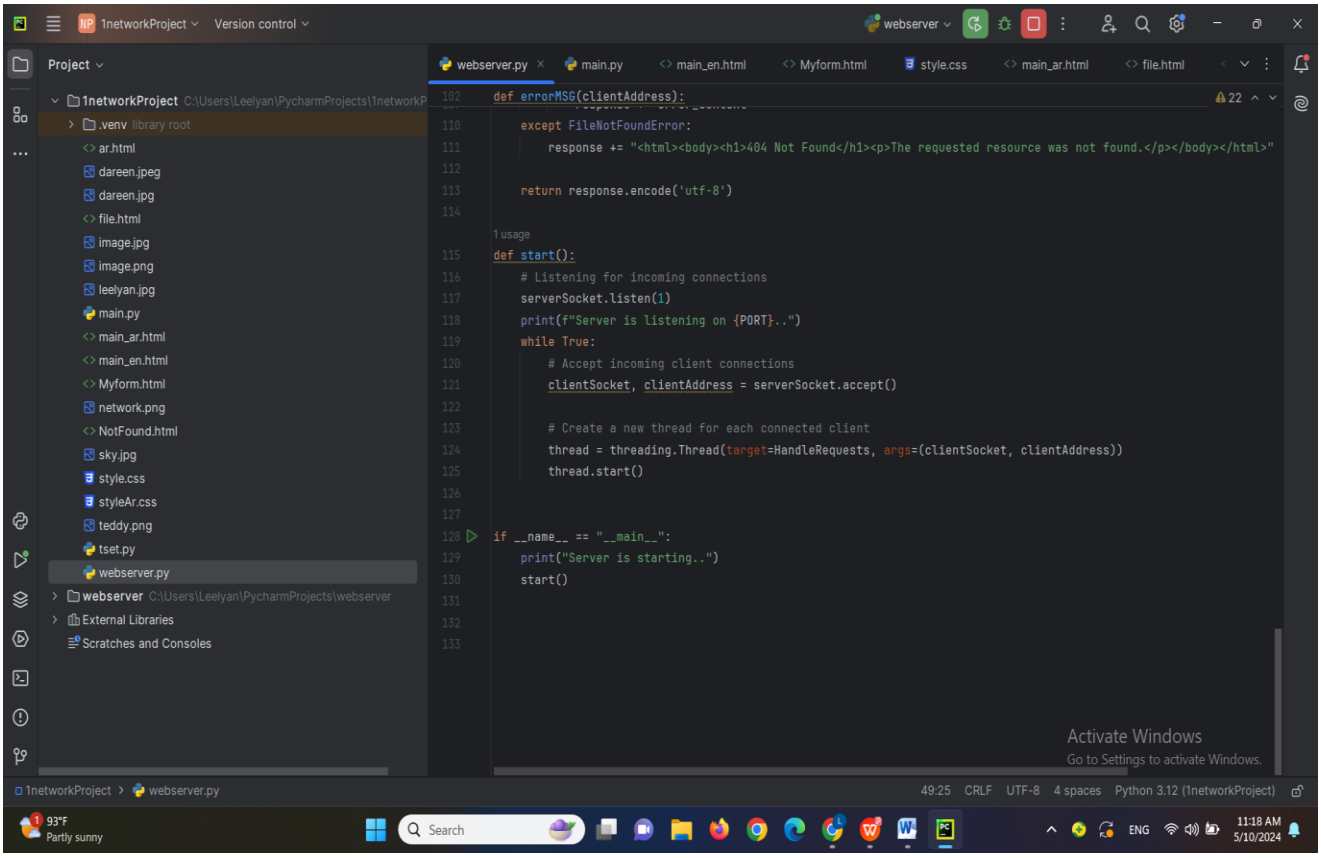
Figure 10:web server






1networkProject
Version control
webserver
Project
webserver.py
main.py
main_en.html
Myform.html
style.css
main_ar.html
file.html
1networkProject C:\Users\Leelyan\PycharmProjects\1networkP
.venv library root
ar.html
dareen.jpeg
dareen.jpg
file.html
image.jpg
image.png
leelyan.jpg
main.py
main_ar.html
main_en.html
Myform.html
network.png
NotFound.html
sky.jpg
style.css
styleAr.css
teddy.png
tset.py
webserver.py
webserver C:\Users\Leelyan\PycharmProjects\webserver
External Libraries
Scratches and Consoles
def ProcessRequests(request,clientAddress):
# Check if the method is GET
if request.startswith("GET /"):
# Extract the path from the request
path = request.split()[1]
# Print the received HTTP request to the terminal
print(f"Received HTTP request: {request}")
# Send main_en.html file with Content-Type: text/html
if path == '/' or path == '/index.html' or path == '/main_en.html' or path == '/en':
return Serve(path: 'main_en.html', type: 'text/html')
# Response with main_ar.html which is an Arabic version of main_en.html
elif path == '/ar':
return Serve(path: 'main_ar.html', type: 'text/html')
# Send the requested html file with Content-Type: text/html
elif path.endswith('.html'):
return Serve(path[1:], type: 'text/html')
# Send the requested css file with Content-Type: text/css
elif path.endswith('.css'):
return Serve(path[1:], type: 'text/css')
# Send the png image with Content-Type: image/png
elif path.endswith('.png'):
return Serve(path[1:], type: 'image/png')
Send the jpg image with Content-Type: image/jpeg
Activate Windows
Go to Settings to activate Windows.
1networkProject > webserver.py
49:25 CRLF UTF-8 4 spaces Python 3.12 (1networkProject)
93°F
Partly sunny
Search
ENG
11:17 AM
5/10/2024


1networkProject
Version control
webserver
Project
webserver.py
main.py
main_en.html
Myform.html
style.css
main_ar.html
file.html
1networkProject C:\Users\Leelyan\PycharmProjects\1networkP
.venv library root
ar.html
dareen.jpeg
dareen.jpg
file.html
image.jpg
image.png
leelyan.jpg
main.py
main_ar.html
main_en.html
Myform.html
network.png
NotFound.html
sky.jpg
style.css
styleAr.css
teddy.png
tset.py
webserver.py
webserver C:\Users\Leelyan\PycharmProjects\webserver
External Libraries
Scratches and Consoles
def ProcessRequests(request,clientAddress):
# Send the jpg image with Content-Type: image/jpeg
elif path.endswith('.jpg'):
return Serve(path[1:], type: 'image/jpeg')
# Redirect to stackoverflow.com website
elif path == '/so':
return "HTTP/1.1 307 Temporary Redirect\nLocation: https://stackoverflow.com\n\n".encode('utf-8')
# Redirect to itc website
elif path == '/itc':
return "HTTP/1.1 307 Temporary Redirect\nLocation: https://itc.birzeit.edu/\n\n".encode('utf-8')
Default: 404 Not Found
return errorMSG(clientAddress)
1 usage
def errorMSG(clientAddress):
response = "HTTP/1.1 404 Not Found\nContent-Type: text/html\n\n"
try:
with open("NotFound.html", "r") as errorFile:
error_content = errorFile.read()
error_content = error_content.replace(old: "[Client IP address and Port]", new: f"{clientAddress[0]}
response += error_content
except FileNotFoundError:
response += "<html><body><h1>404 Not Found</h1> The requested resource was not found. </body></html>"
return response.encode('utf-8')
1 usage
Activate Windows
Go to Settings to activate Windows.
1networkProject > webserver.py
49:25 CRLF UTF-8 4 spaces Python 3.12 (1networkProject)
93°F
Partly sunny
Search
ENG
11:18 AM
5/10/2024



- **Explanation :**

  - ✓ **Importing Modules:** The code imports the necessary modules socket and threading.
  - ✓ **Defining Host and Port:** It sets the host to the local machine's IP address and the port to 6060.
  - ✓ **Creating and Binding Socket:** It creates a TCP socket and binds it to the specified host and port.
  - ✓ **Function to Serve HTML Files (Serve()):** Reads and serves HTML files, returning appropriate HTTP responses.
  - ✓ **Function to Handle Client Requests (HandleRequests()):** Receives client requests, processes them, and sends responses accordingly.

- ✓ **Function to Process Client Requests (ProcessRequests()):** Processes HTTP requests based on the requested path, serving files or handling redirection.
- ✓ **Function to Handle Error Messages (errorMSG()):** Constructs HTTP responses for "404 Not Found" errors.
- ✓ **Main Server Loop (start()):** Listens for incoming connections, accepts them, and handles each connection in a separate thread.
- ✓ **Main Execution Block:** Starts the server by calling the start() function.

This code essentially creates a basic HTTP server that serves files and handles client requests, all while using multithreading to handle multiple clients simultaneously.

---

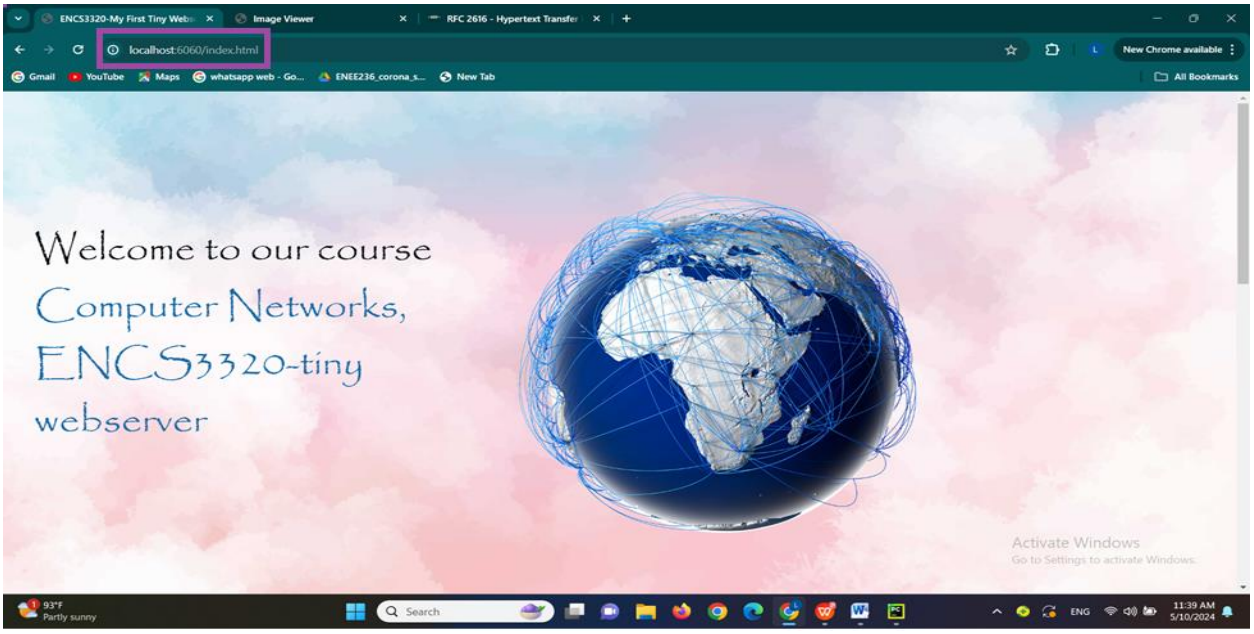
0. From rfce2616, what is Entity Tag Cache Validators in the HTTP protocol and why do we need it?

The ETag response-header field value, an entity tag, provides for an "opaque" cache validator. This might allow more reliable validation in situations where it is inconvenient to store modification dates, where the one-second resolution of HTTP date values is not sufficient, or where the origin server wishes to avoid certain paradoxes that might arise from the use of modification dates

---

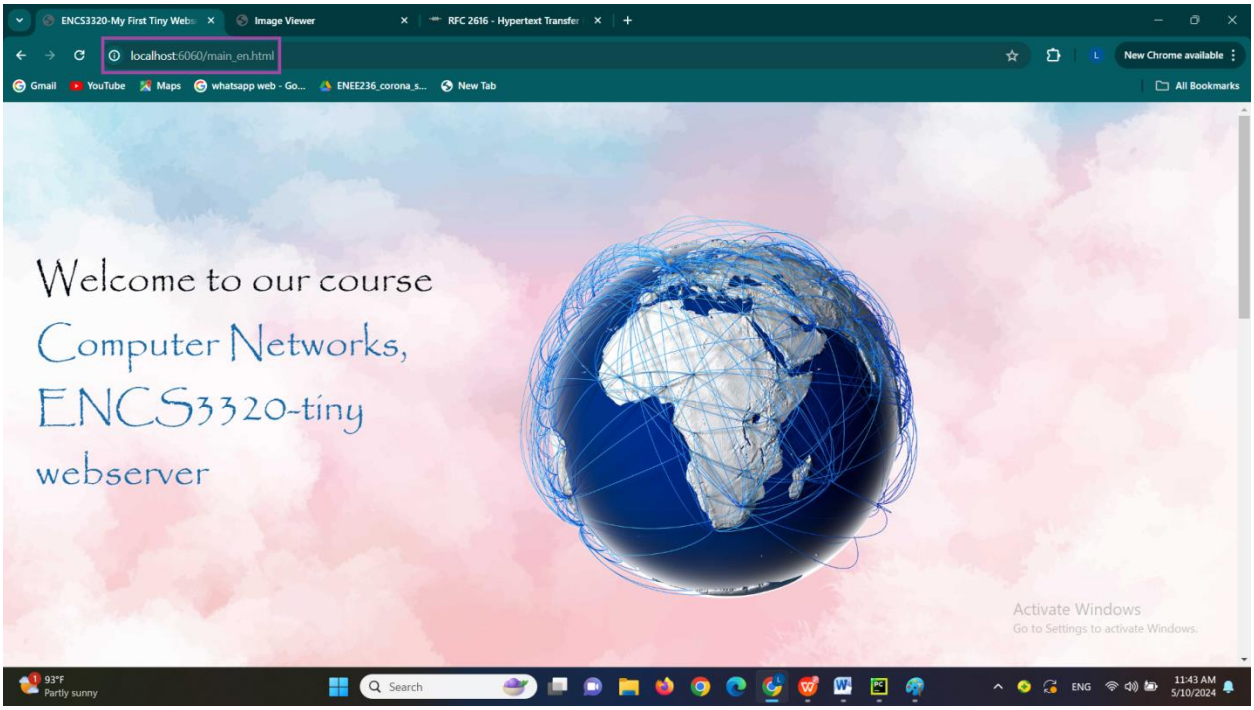
if the request is / or /index.html or /main_en.html or /en (for example localhost:6060/ or localhost:6060/en) then the server should send main_en.html file with Content-Type: text/html.

Figure 11:The HTTP request '/index.html'



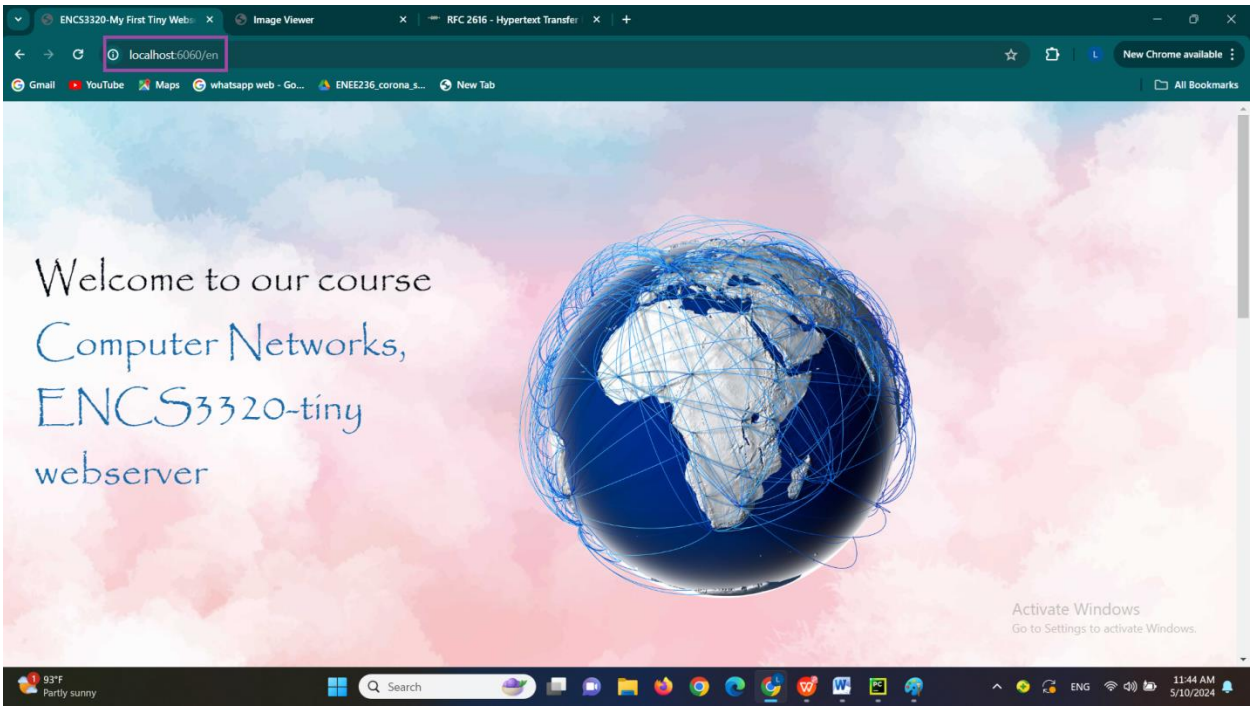
The HTTP request '/index.html' redirects the client to our webpage.

Figure 12: The HTTP request '/main_en.html'



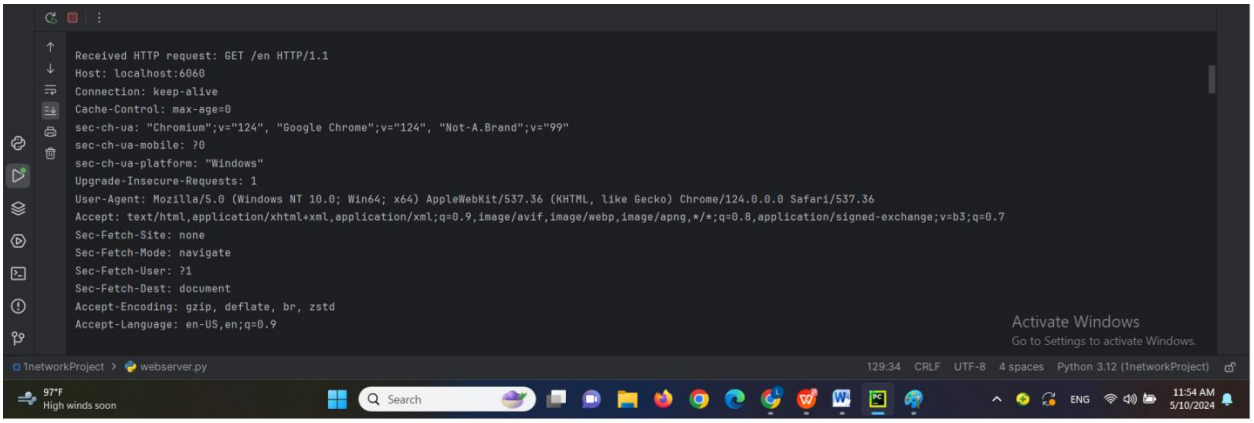
The HTTP request '/main_en.html' redirects the client to our webpage.

Figure 13:The HTTP request '/en'



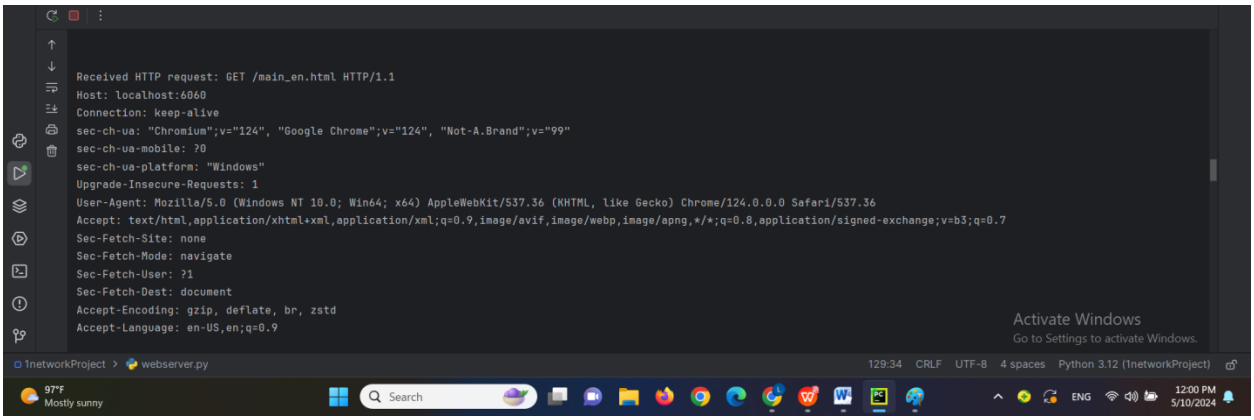
The HTTP request '/en' redirects the client to our webpage.

Figure 14:HTTP request for '/en'



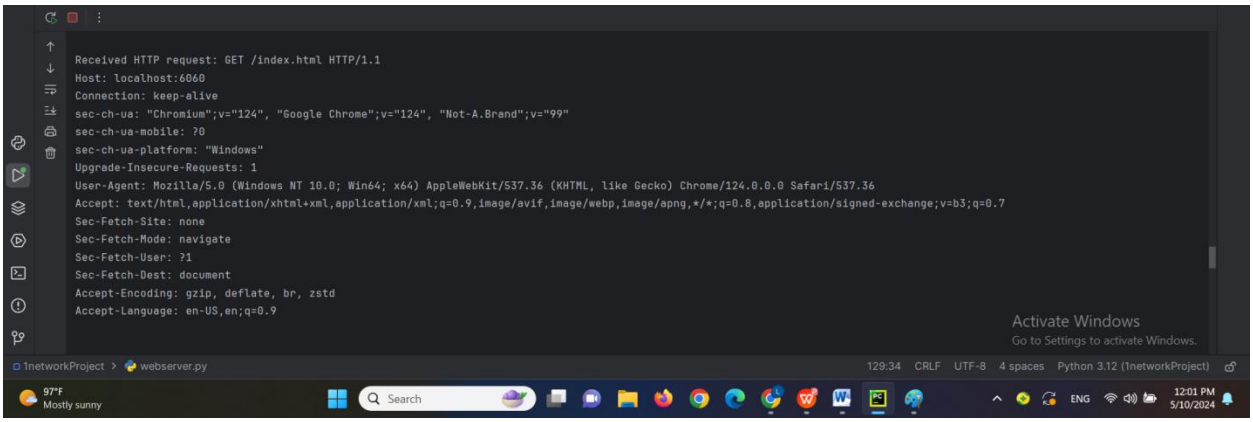
HTTP request for **'/en'.**

Figure 15:HTTP request for '/main_en.html'



HTTP request for **'/main_en.html'.**

Figure 16:HTTP request for '/index.html'



HTTP request for **'/index.html'.**

- **Explanation:**

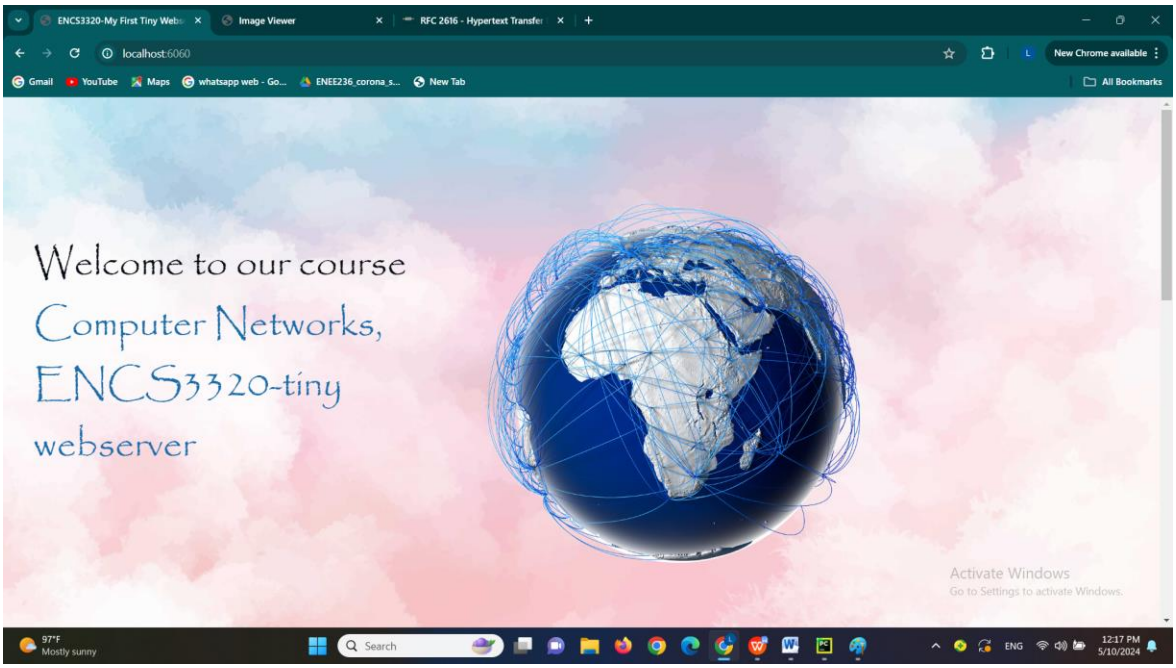
  The previous HTTP request messages: (taking main_en.html as an example)

  The request line: "GET /main_en.html HTTP/1.1".

  - ✓ **GET:** The method for sending data to server.
  - ✓ **/main_en.html:** The URL of the HTML file requested by the client.
  - ✓ **HTTP/1.1:** The version of the protocol used by the client.
  - ✓ **The next line** the port number: (6060)
  - ✓ **The header lines:** the rest of the request message are header lines each with a header field name and a value.
  - ✓ **The body:** the body is empty.

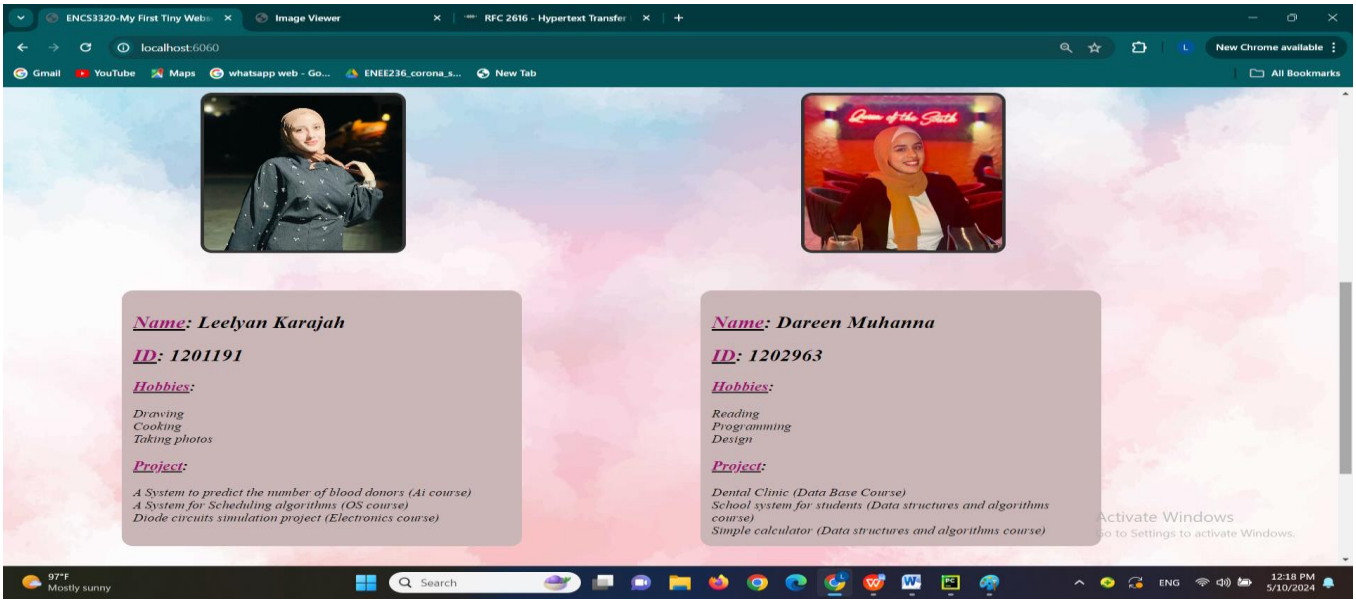
- **The main_en.html file should contain HTML webpage that contains :**

  a. "ENCS3320-My First Tiny Webserver" in the title.
  b. "Welcome to our course **Computer Networks, ENCS3320-tiny webserver"** (part of the phrase is in **Blue**).

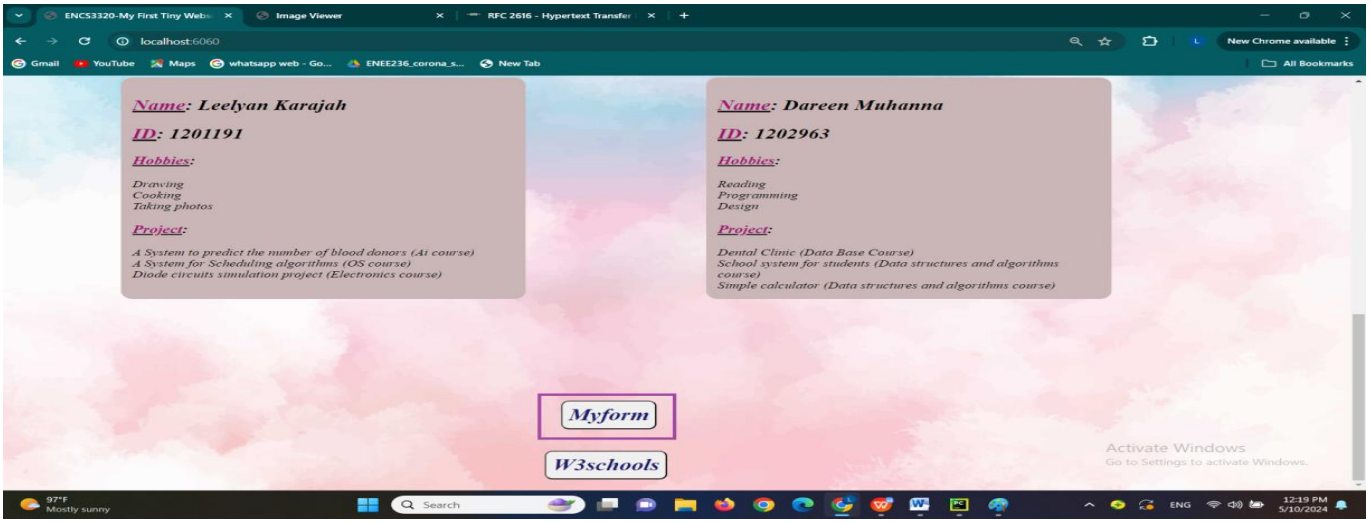


  c. Group members names and IDs
  d. Some information about the group members. For instance, projects you have done during different course (programming, electrical, math, etc), skills, hobbies, etc.

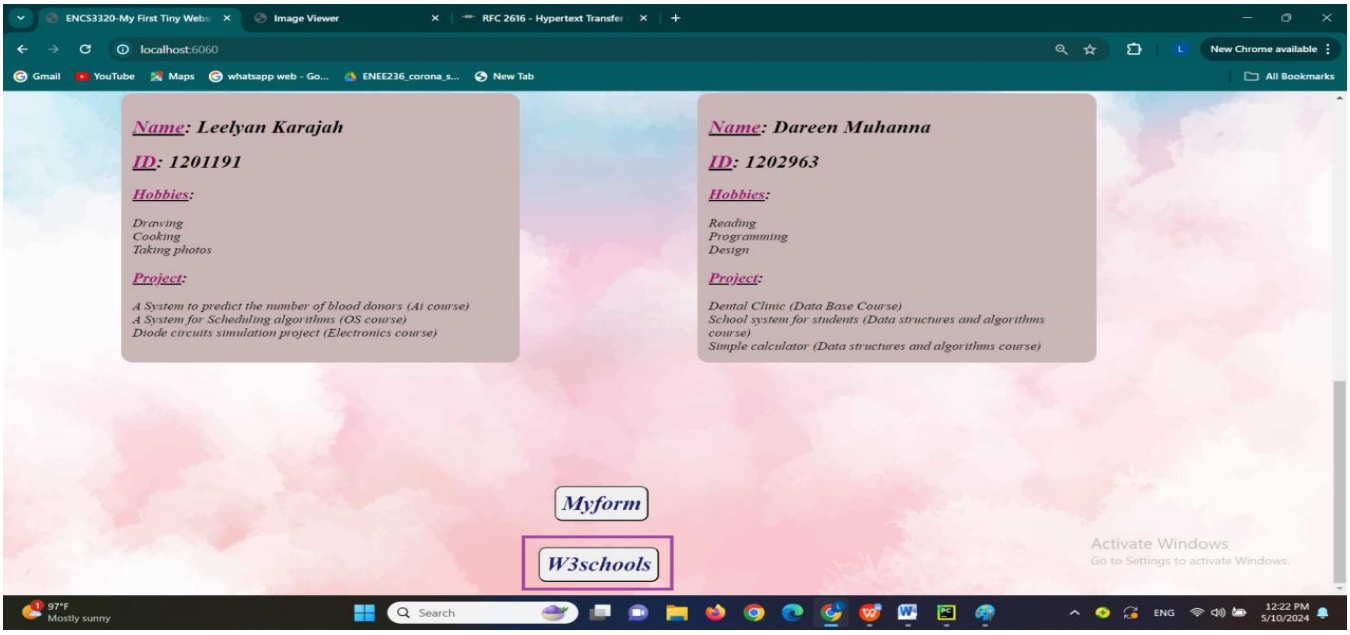
e. Use CSS to make the page looks nice
f. Divide the page in different boxes and put student's information in the different boxes
g. Include CSS as a separate file
h. The page should contain at least An image with extention.jpg and an image with extension .png



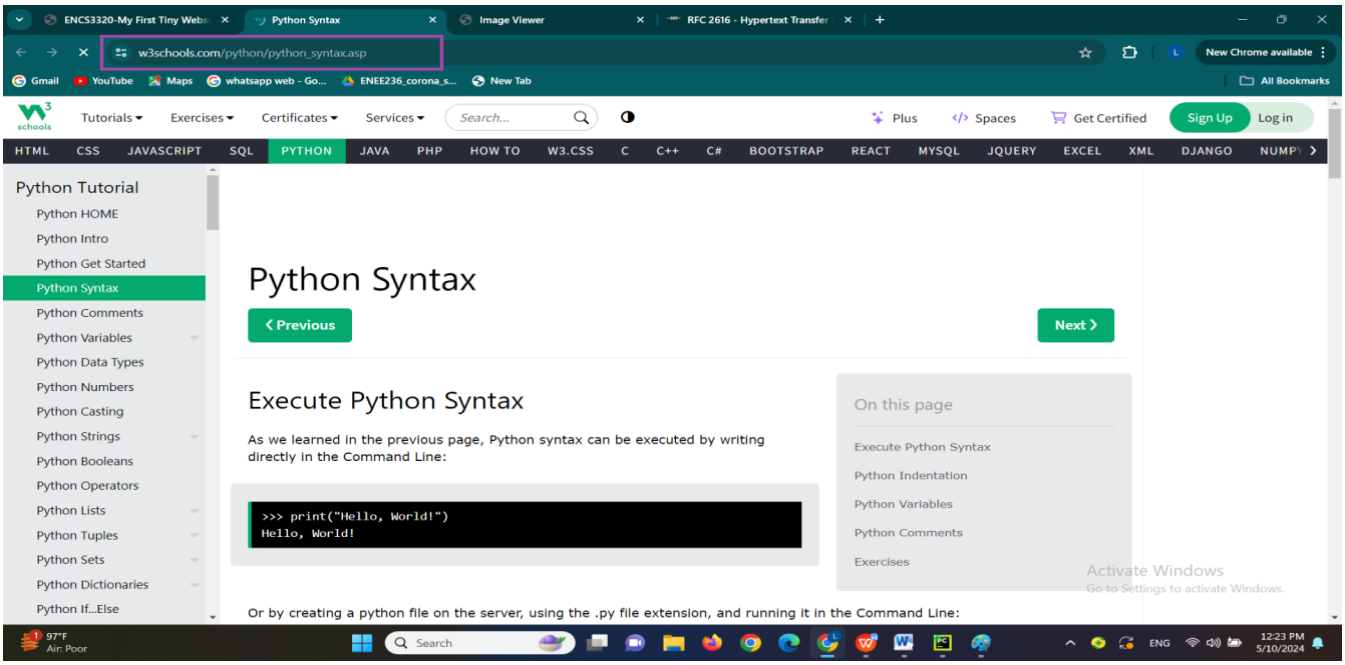
i. A link to a local html file (myform.html)



j. a link to https://www.w3schools.com/python/python_syntax.asp

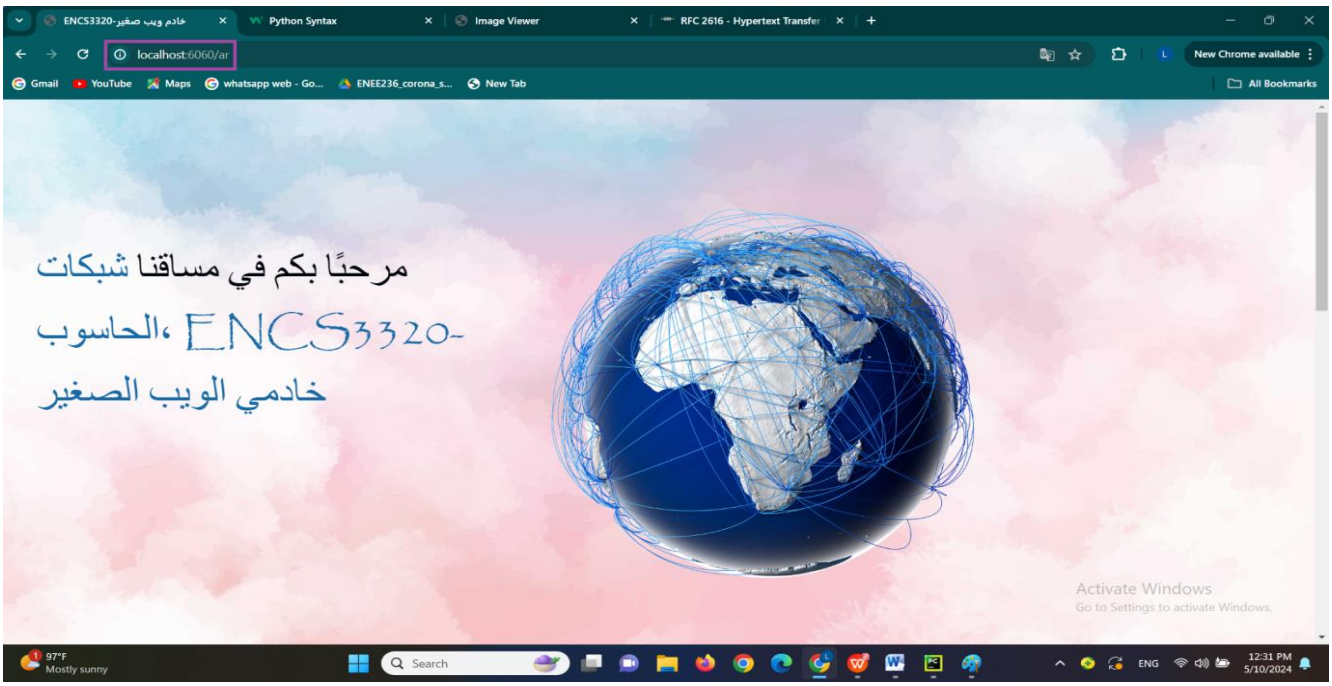


When you click the button it will take you to https://www.w3schools.com/python/python_syntax.asp.

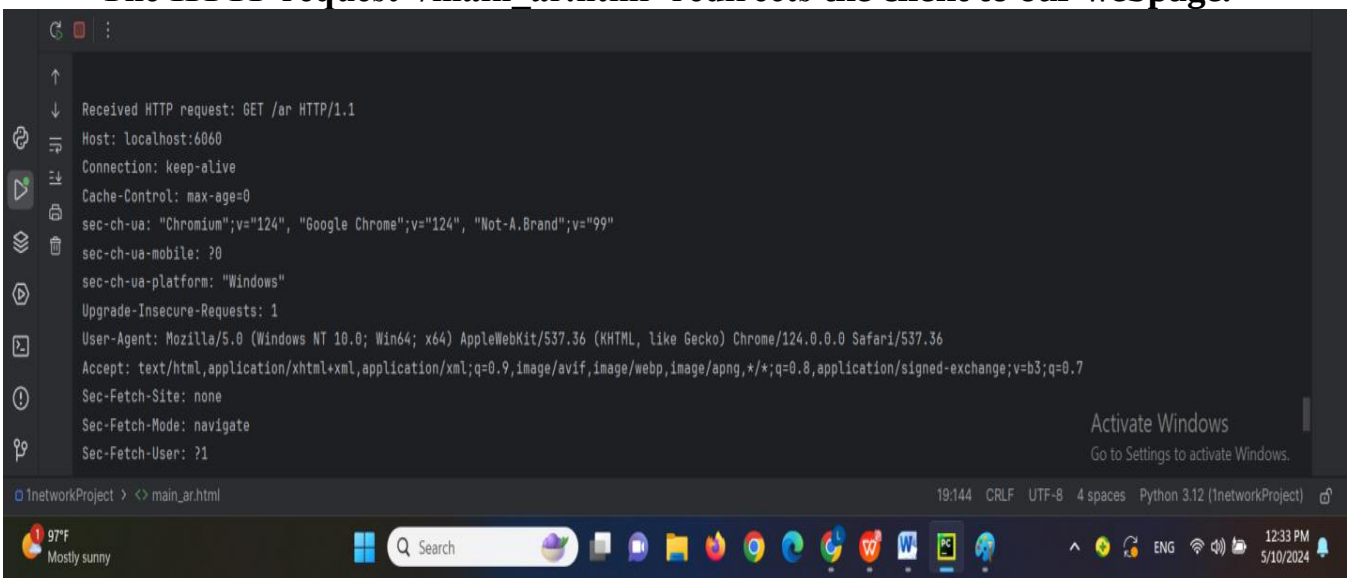


1. If the request is /ar then the server response with main_ar.html which is an Arabic version of main_en.html

Figure 17:The HTTP request '/main_ar.html'

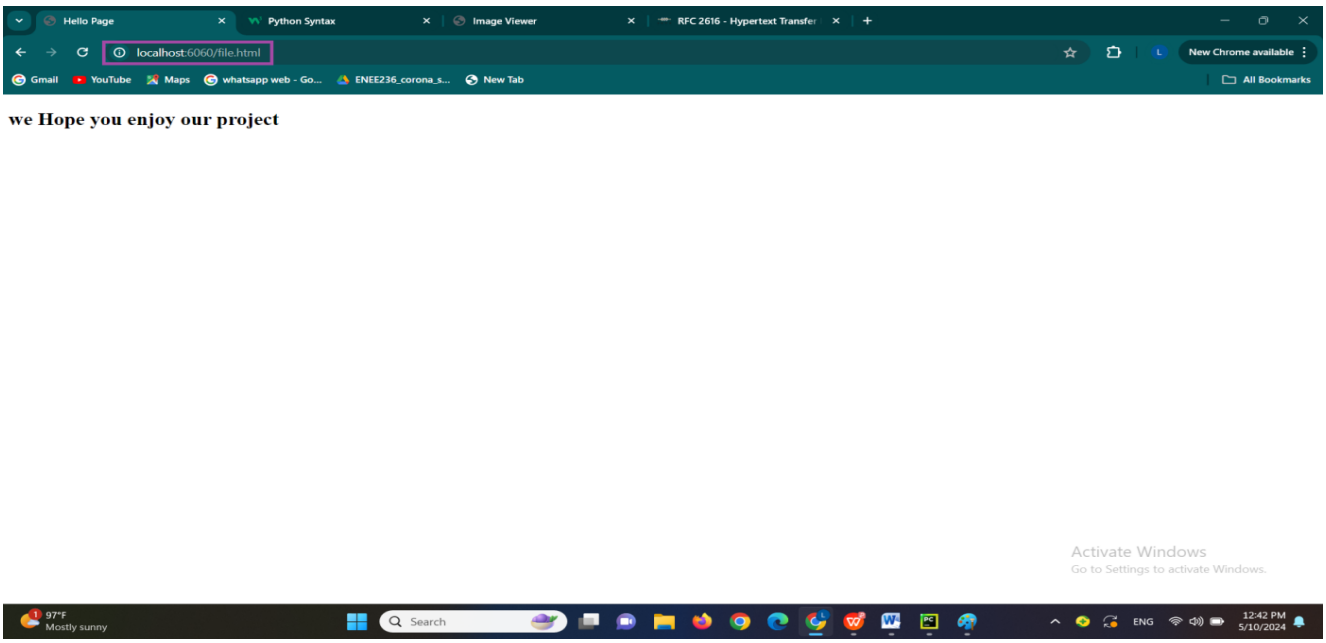


The HTTP request '/main_ar.html' redirects the client to our webpage.

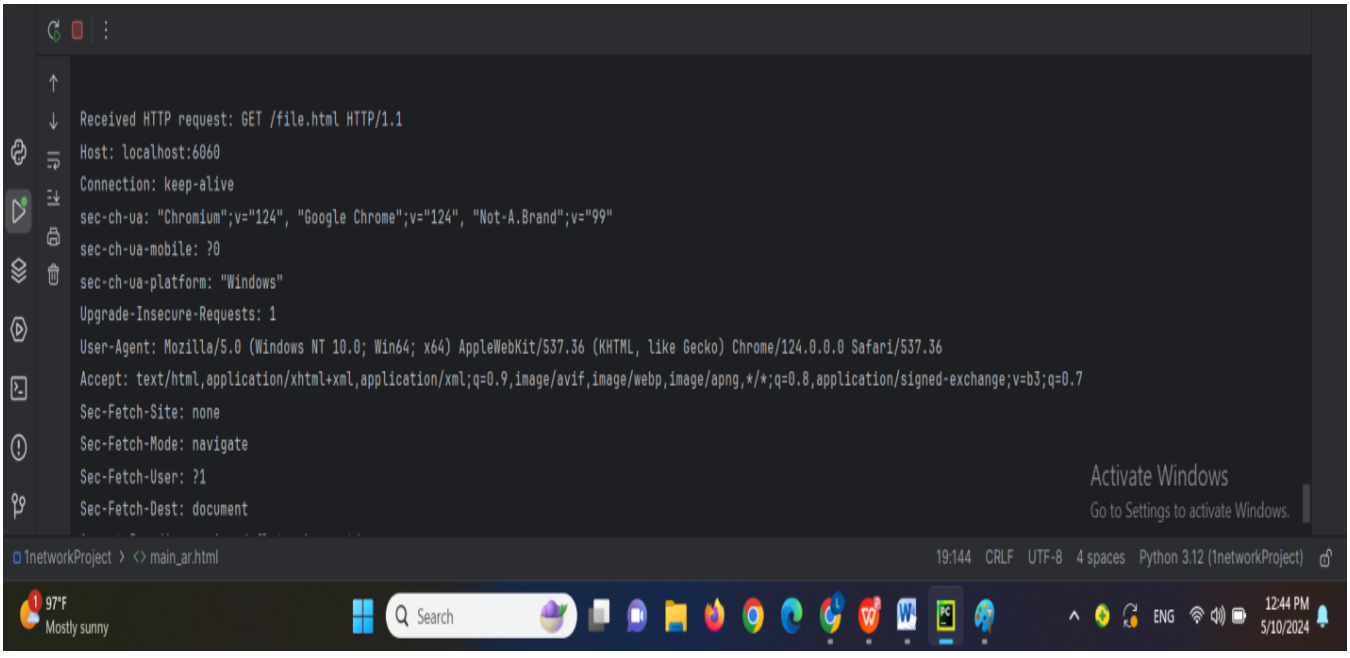


2. if the request is an .html file then the server should send the requested html file with Content-Type: text/html. You can use any html file. Make it general (not only for specific filename)

Figure 18:The HTTP request '/file.html' redirects the client to a webpage designed by html.

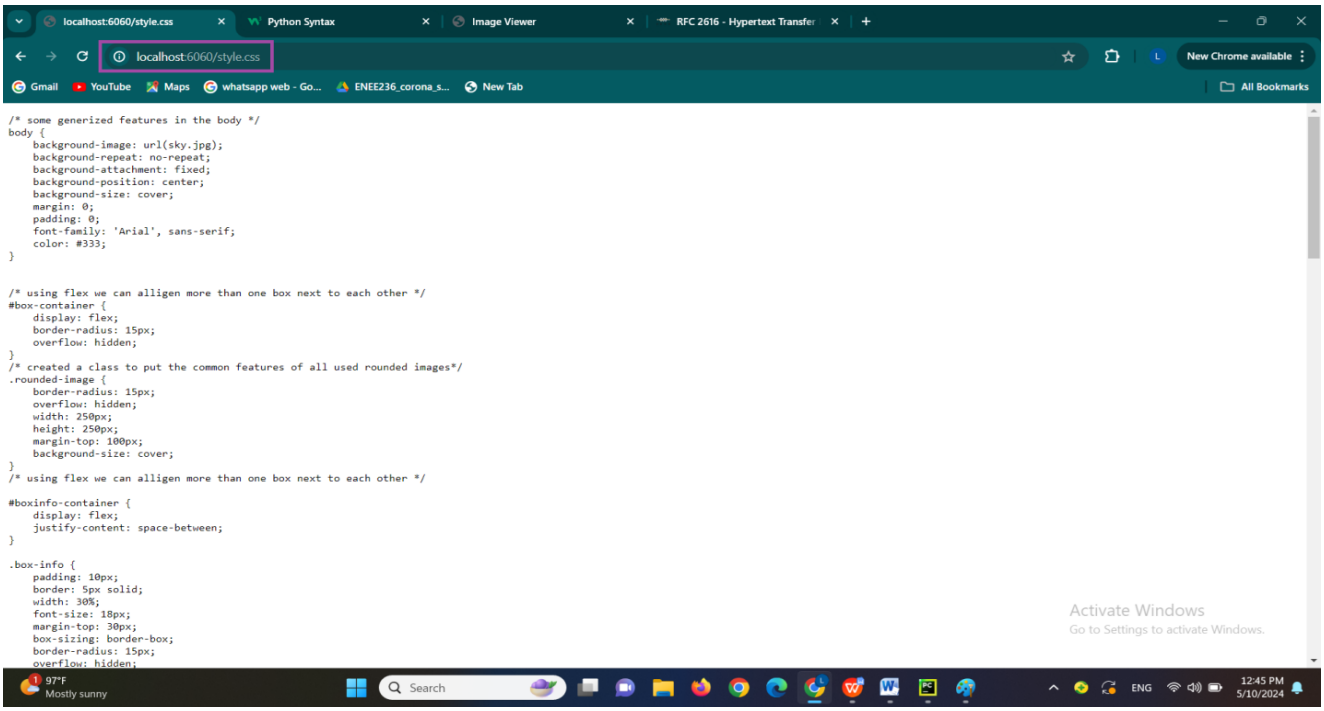


The HTTP request '/file.html' redirects the client to a webpage designed by html.

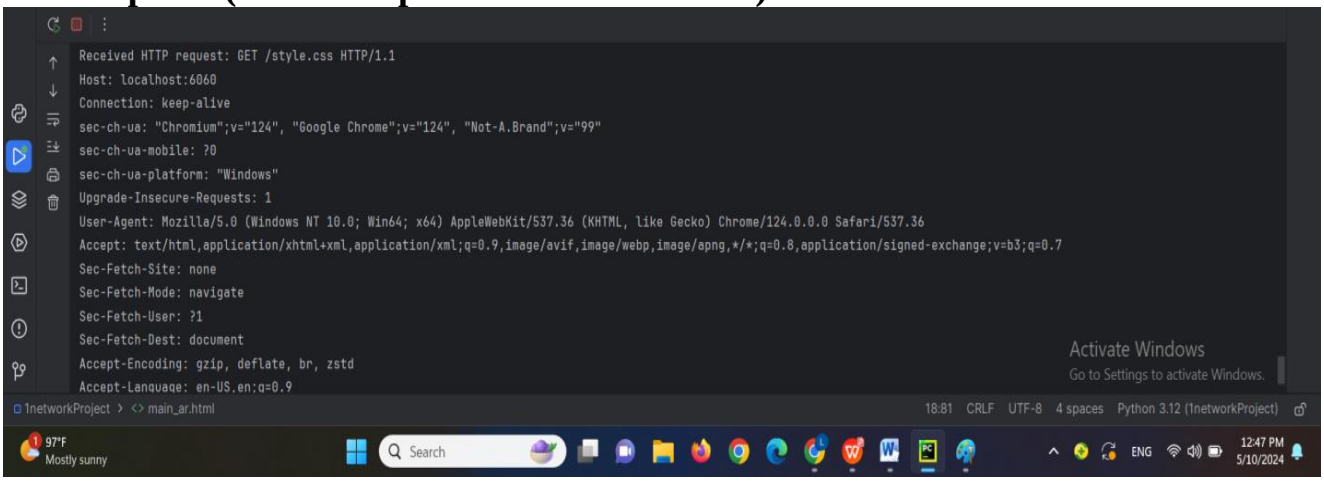


4. if the request is a .css file then the server should send the requested css file with Content-Type: text/css. You can use any CSS file. Make it general (not only for specific filename)

Figure 19:The HTTP request '/style.css' redirects the client to the css file stored on the computer (The same path as the web server).

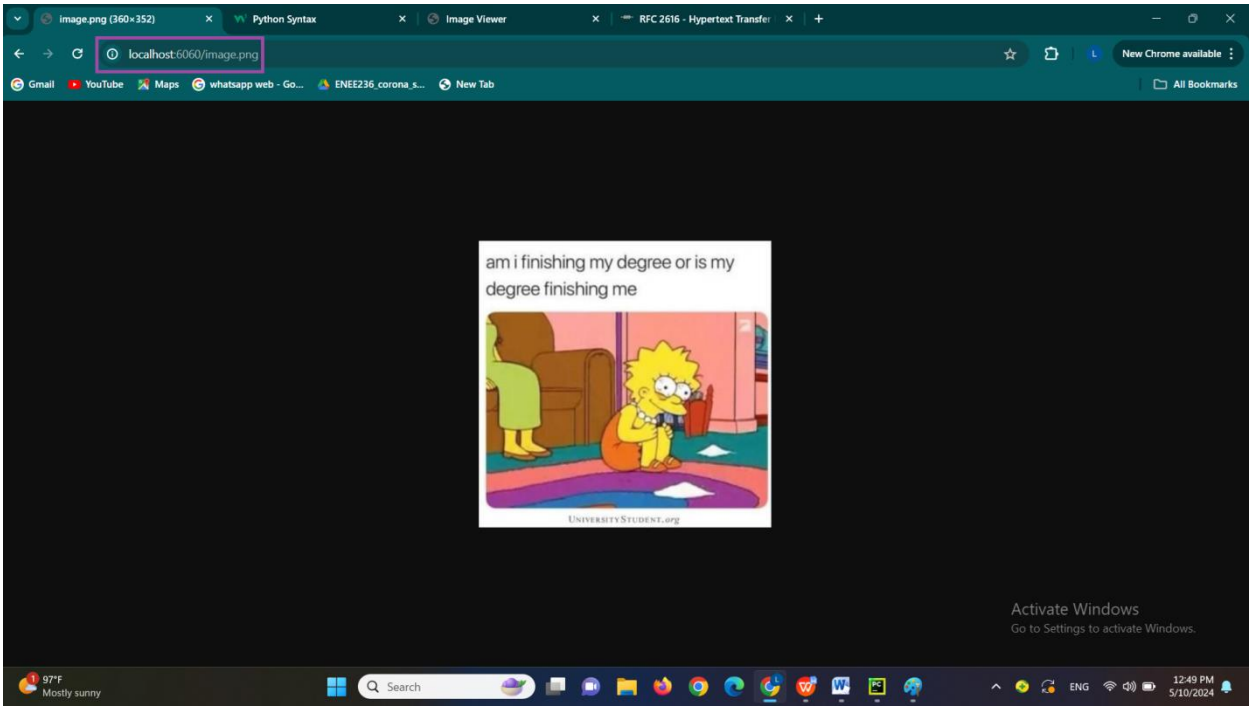


The HTTP request '/style.css' redirects the client to the css file stored on the computer (The same path as the web server).

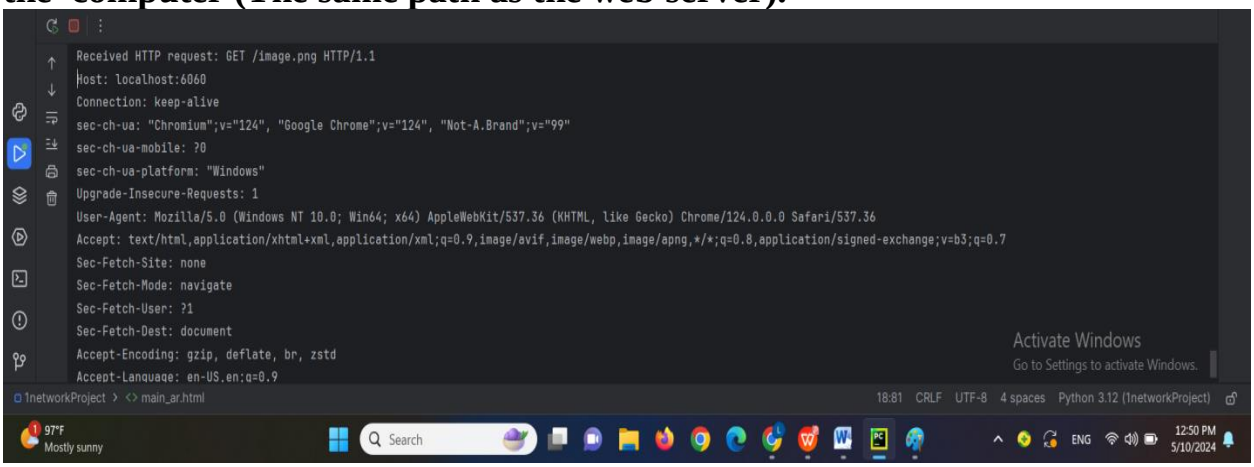


5. if the request is a .png then the server should send the png image with Content-Type: image/png. You can use any image. Make it general (not only for specific filename)

Figure 20:The HTTP request '/image.png' redirects the client to a png image stored on the computer (The same path as the web server).

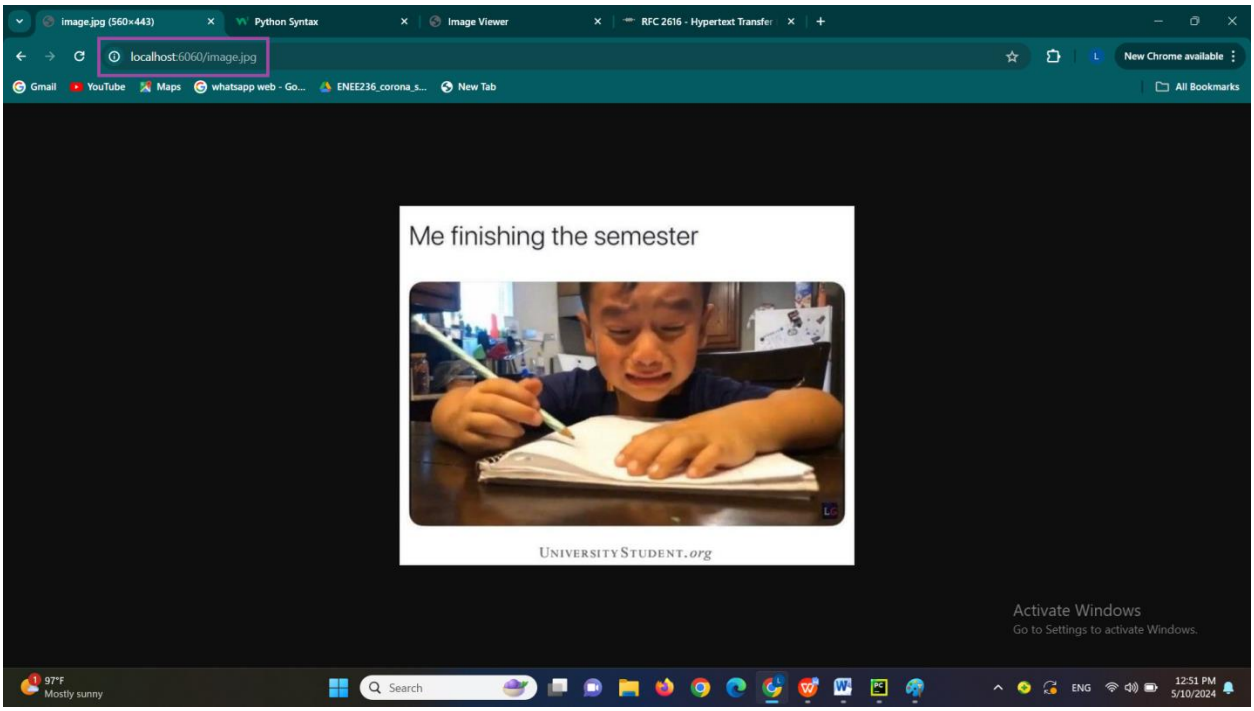


The HTTP request '/image.png' redirects the client to a png image stored on the computer (The same path as the web server).

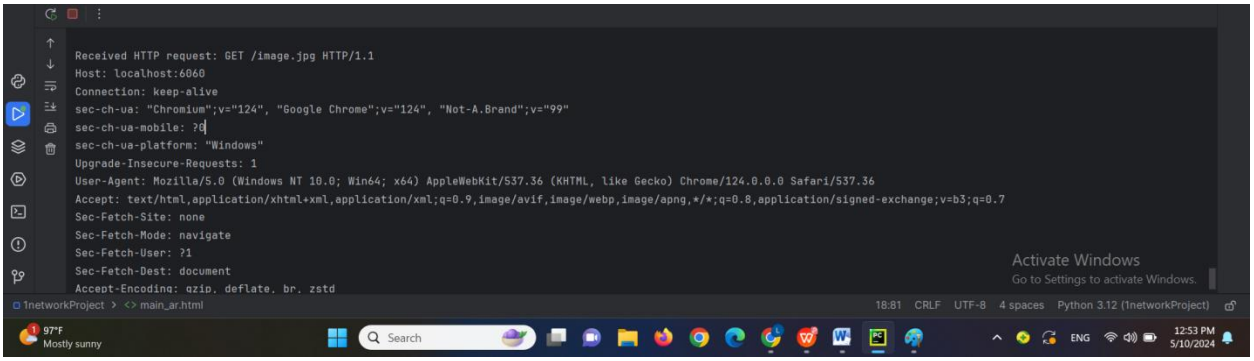


6. if the request is a .jpg then the server should send the jpg image with Content-Type: image/jpeg. You can use any image. Make it general (not only for specific filename)

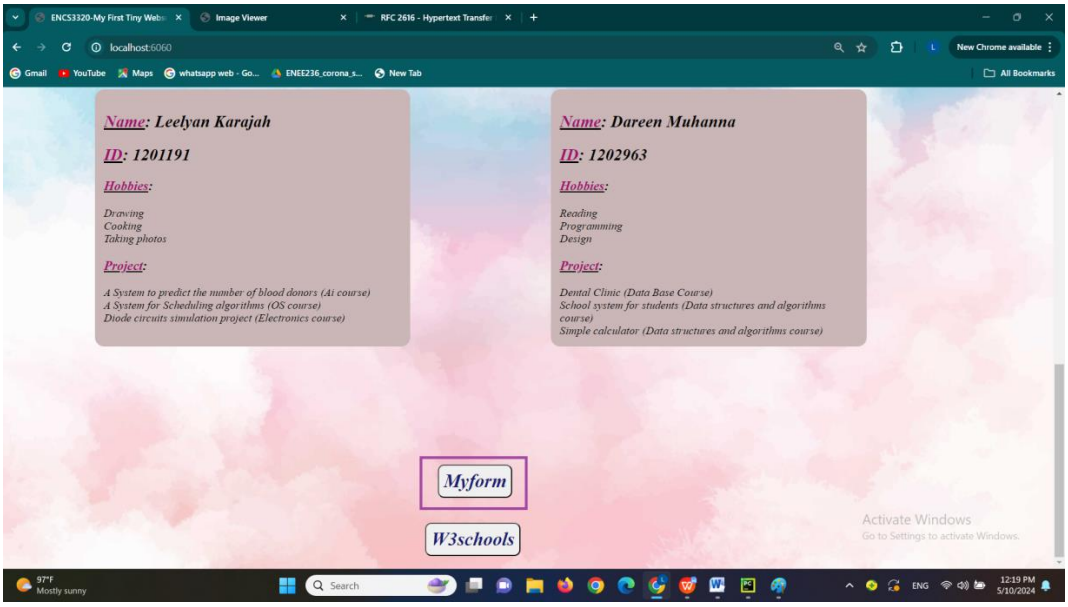
Figure 21:The HTTP request '/image.jpg' redirects the client to a jpg image stored on the computer (The same path as the web server).

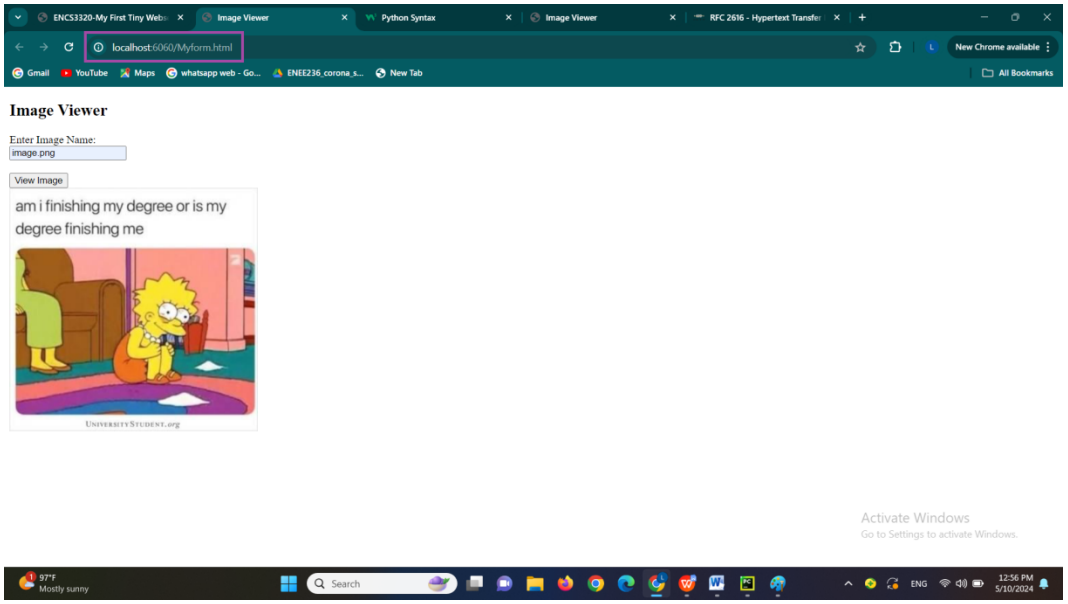


The HTTP request '/image.jpg' redirects the client to a jpg image stored on the computer (The same path as the web server).



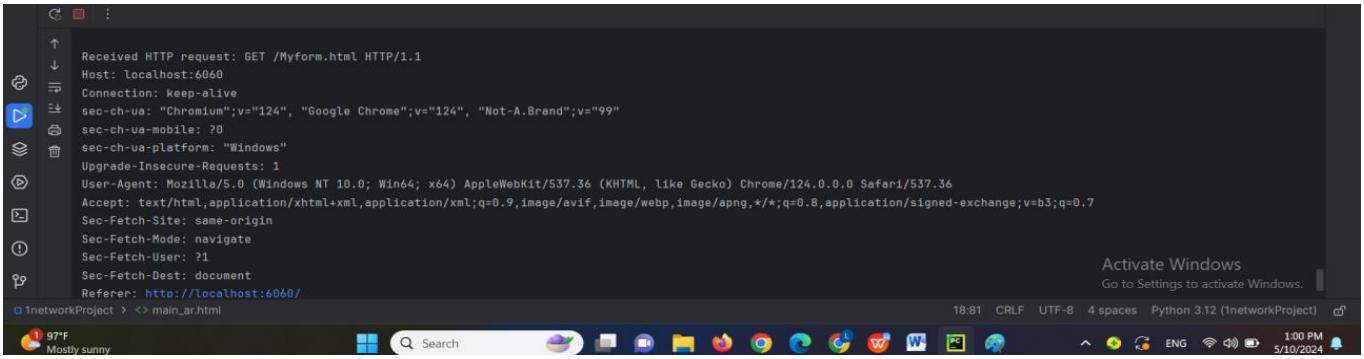
7. Use myform.html to get image by typing the name of the image in a box.





The HTTP request '/Myform.html'

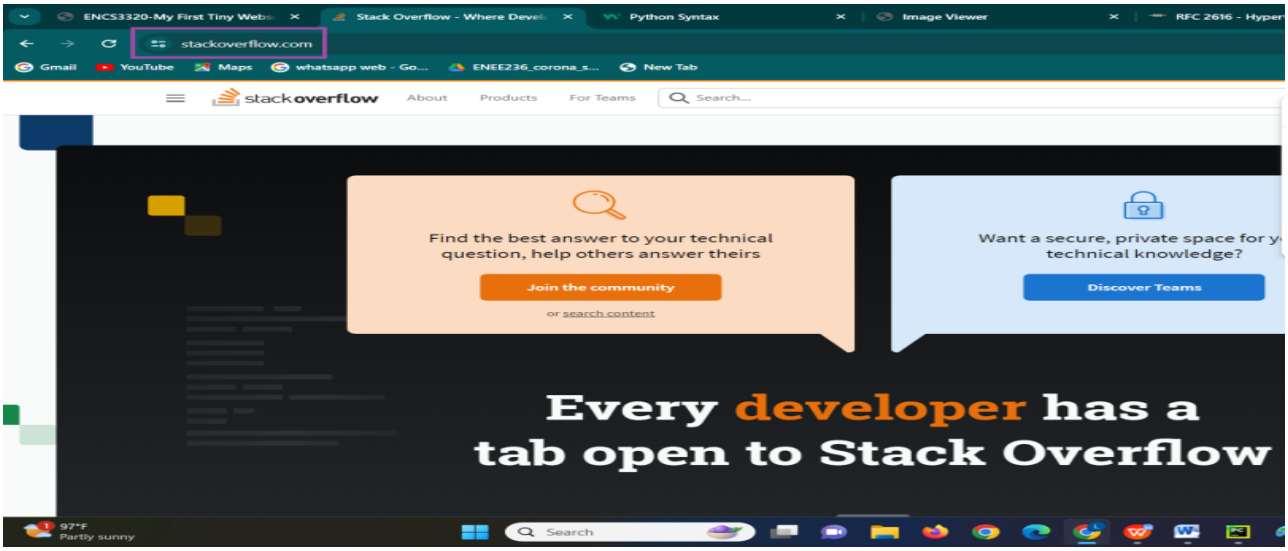
Figure 22:The HTTP request '/Myform.html'

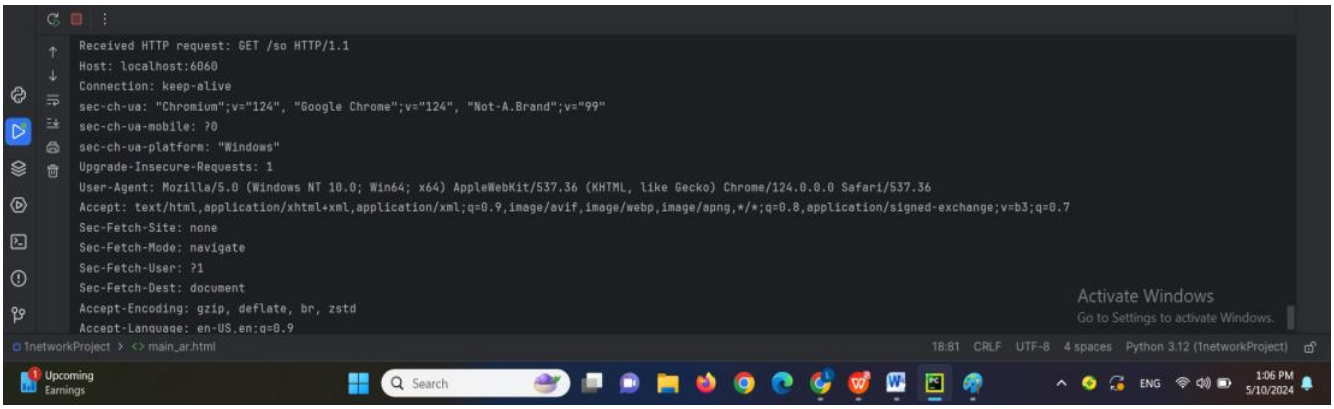


8. Use the status code 307 Temporary Redirect to redirect the following

- If the request is /so then redirect to stackoverflow.com website

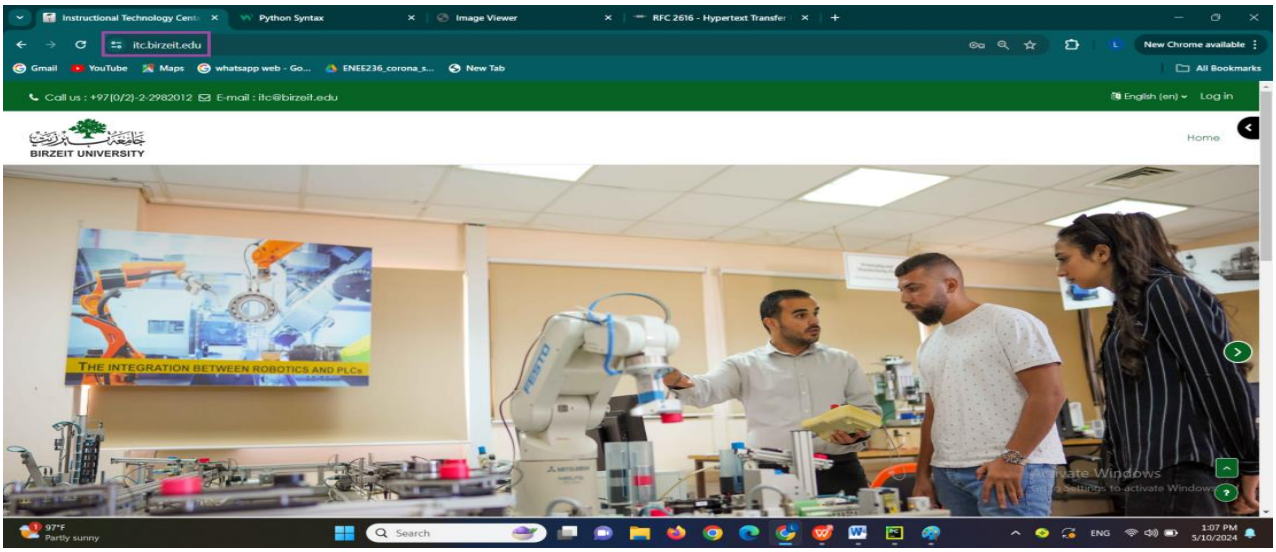
Figure 23:stackoverflow.com website

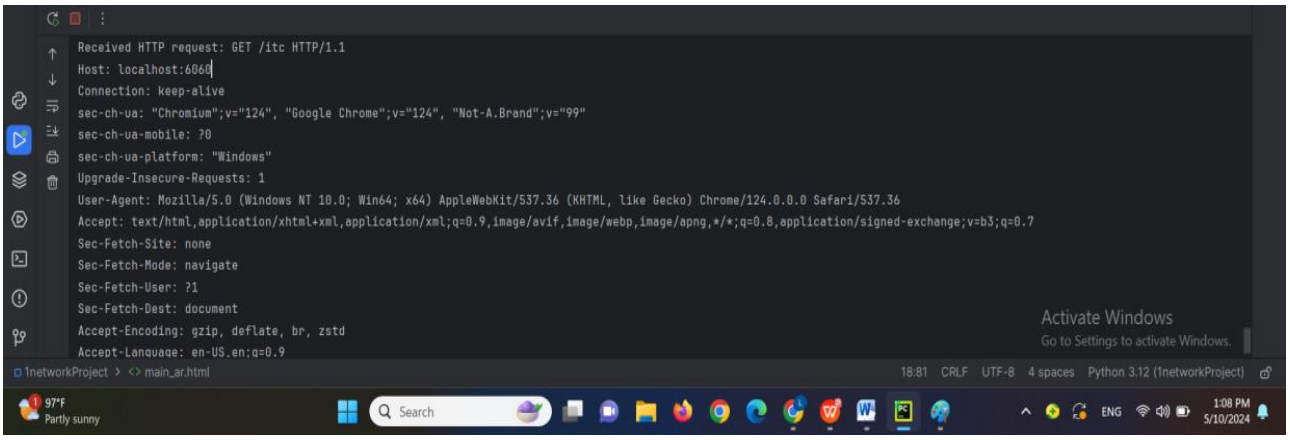




❖ If the request is /itc then redirect to itc.birzeit.edu website

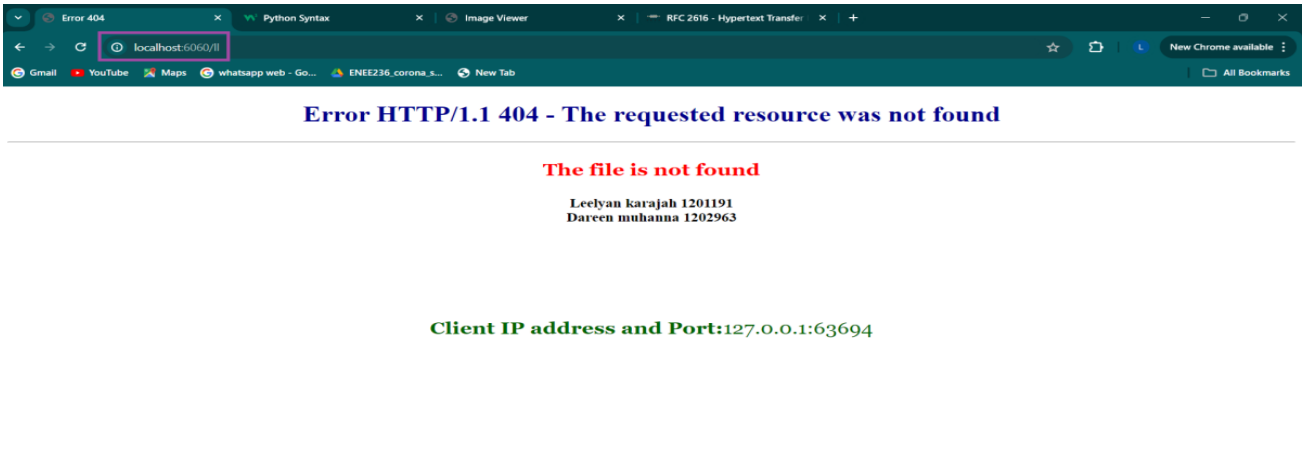
Figure 24:itc.birzeit.edu website

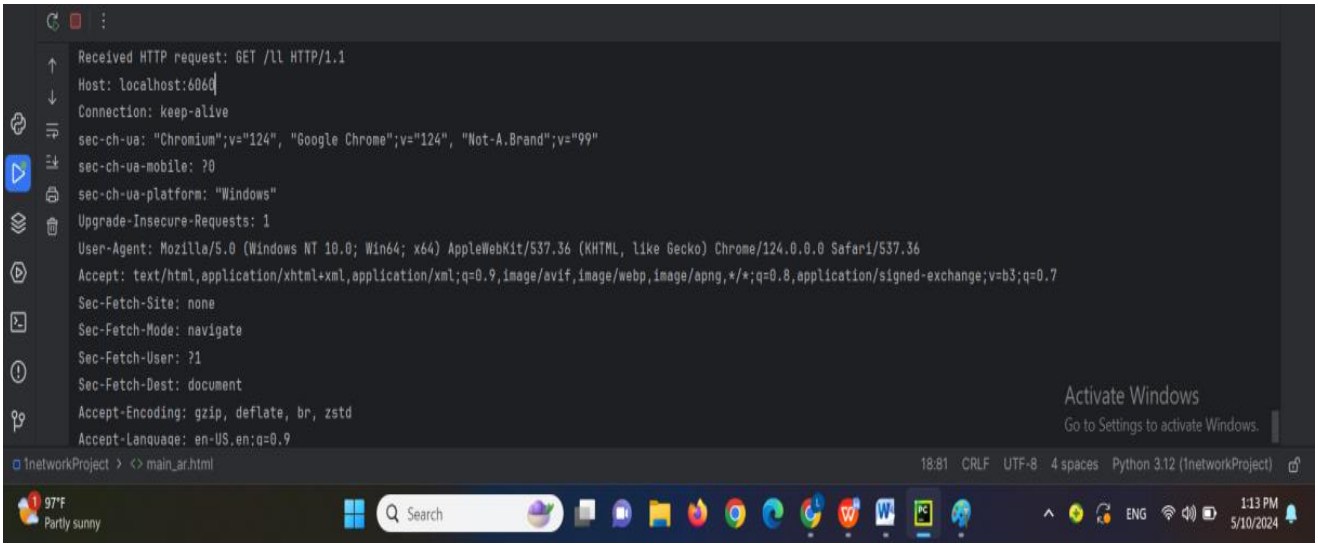




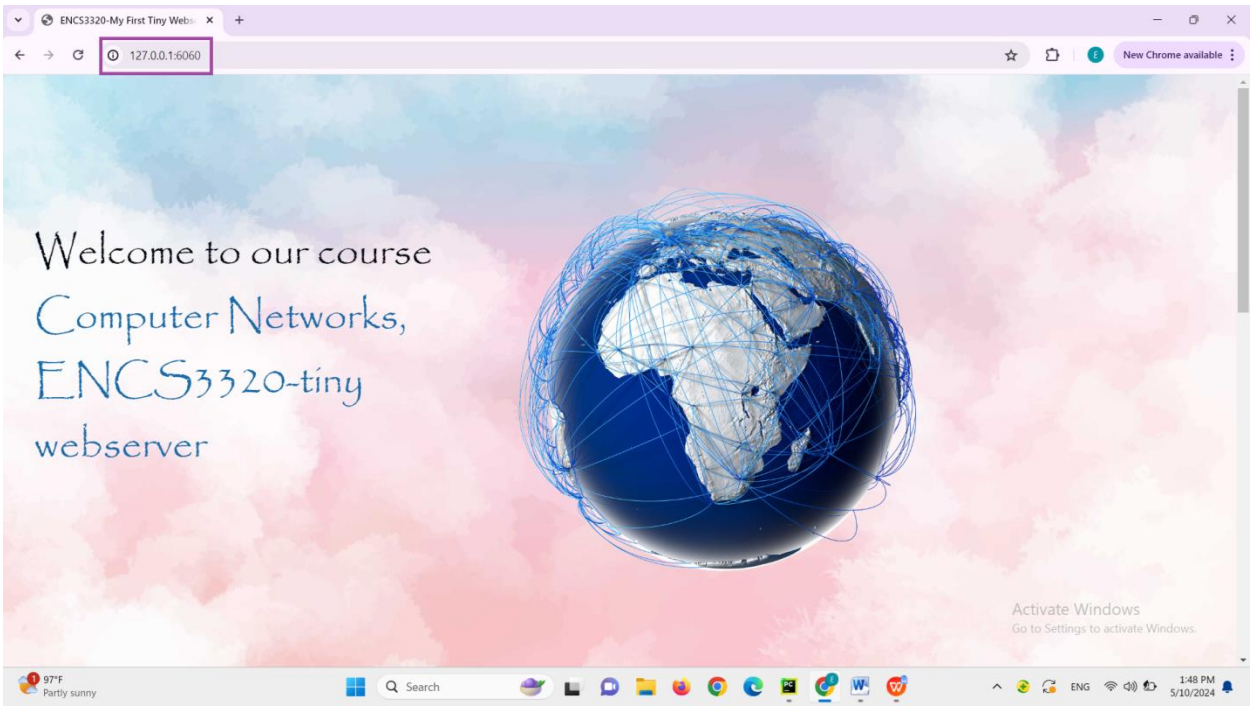
9. If the request is wrong or the file doesn't exist the server should return a simple HTML webpage that contains (Content-Type: text/html)

1. "HTTP/1.1 404 Not Found" in the response status
2. "Error 404" in the title
3. "The file is not found" in the body in **red**
4. Your names and IDs in **Bold**
5. The IP and port number of the client





10. Test the project from a browser on the same computer and from a different computer or phone.



Codes

English version

- HTML (main_en.html)

Figure 25:HTML english version

webserver.py | main.py | main_en.html | Myform.html | style.css | main_ar.html | file.html

```
<!DOCTYPE html>
<html>
<head>
    <!--This meta tag defines the character encoding for the HTML document-->
    <meta charset="UTF-8">
    <!--This meta tag is used to control the viewport behavior on different devices-->
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <!--setting the title as it was requested-->
    <title>ENCS3320-My First Tiny Webserver</title>
  <!--this line is very important to connect ove html file to the related css sheet -->
    <link rel="stylesheet" href="style.css">
</head>
<body>
<!--we defined a container to help us align two boxes next to each other as well as to specify the common features of

    <div id="intro-container">
     <!--set a new id for the box0 to be able to control its unique features in css sheet -->

    <div id="box0">
         <!--we used the element span to control the color of specific text  -->

        <span style="color: rgb(1, 6, 15);"> Welcome to our course </span> <span style="color: rgb(24, 106, 168);">Com
    </div>
         <!--set a new id for the box intro to be able to control its unique features in css sheet -->

    <div id="intro">
    </div>
</div>
```

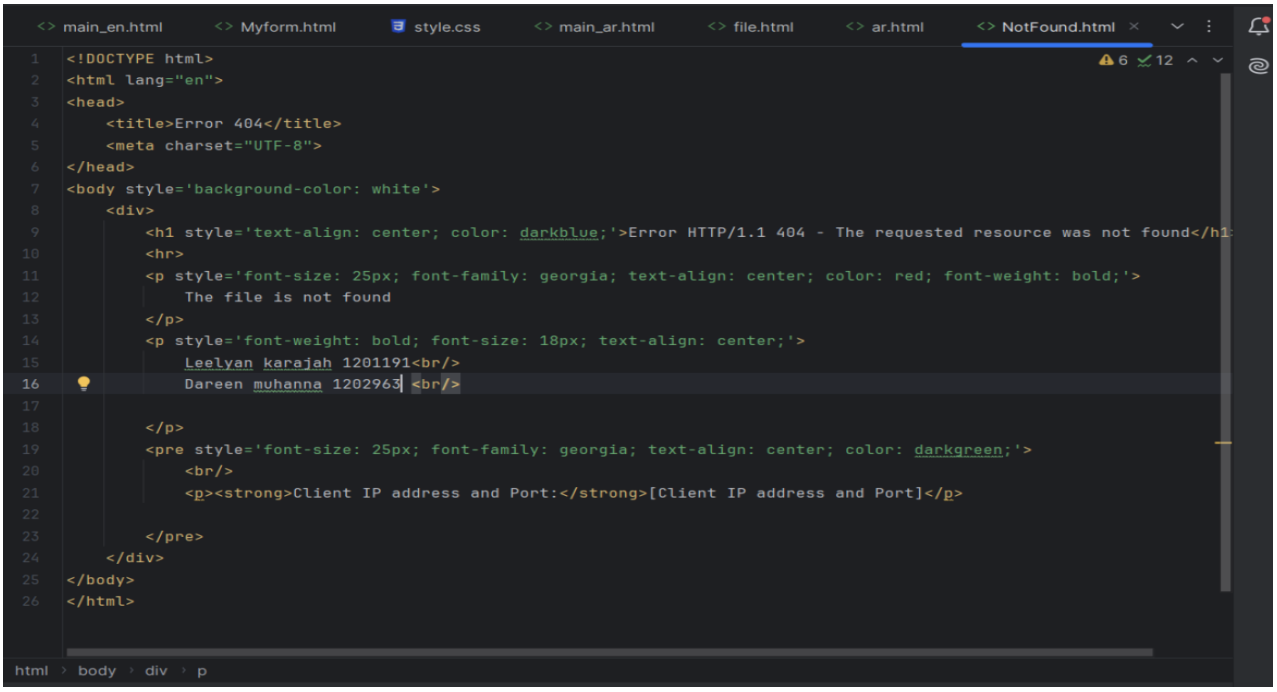
```
 <!--set a new id for each box to be able to control each of their unique features in css sheet -->

    <div id="box-container">

        <div class="rounded-image" id="box2"></div>
        <div class="rounded-image" id="box3"></div>
    </div>


<!--we defined a container to help us align three boxes next to each other as well as to specify the common features o
 <!--set a new id for each box to be able to control each of their unique features in css sheet -->

    <div id="boxinfo-container">
        <div class="box-info" id="boxinfo1">
            <h2><u><span style="color: rgb(161, 16, 115);">Name</span></u>: Leelyan Karajah</h2>
            <h2><u><span style="color: rgb(161, 16, 115);">ID</span></u>: 1201191</h2>
            <h3><u><span style="color: rgb(161, 16, 115);">Hobbies</span></u>:</h3>
            <ul>
                <li>Drawing</li>
                <li>Cooking</li>
                <li>Taking photos</li>
            </ul>
            <h3><u><span style="color: rgb(161, 16, 115);">Project</span></u>:</h3>
            <ol>
                <li>A System to predict the number of blood donors (Ai course)</li>
```

```
                <li>A System for Scheduling algorithms (OS course)</li>
                <li>Diode circuits simulation project (Electronics course)</li>
            </ol>
        </div>
<div class="box-info" id="boxinfo3">
            <h2><u><span style="color: rgb(161, 16, 115);">Name</span></u>: Dareen Muhanna</h2>
            <h2><u><span style="color: rgb(161, 16, 115);">ID</span></u>: 1202963</h2>
            <h3><u><span style="color: rgb(161, 16, 115);">Hobbies</span></u>:</h3>
            <ul>
                <li>Reading</li>
                <li>Programming</li>
                <li>Design</li>
            </ul>
            <h3><u><span style="color: rgb(161, 16, 115);">Project</span></u>:</h3>
            <ol>
                <li>Dental Clinic (Data Base Course)</li>
                <li>School system for students (Data structures and algorithms course)</li>
                <li>Simple calculator (Data structures and algorithms course)</li>
            </ol>
        </div>
    </div>
```

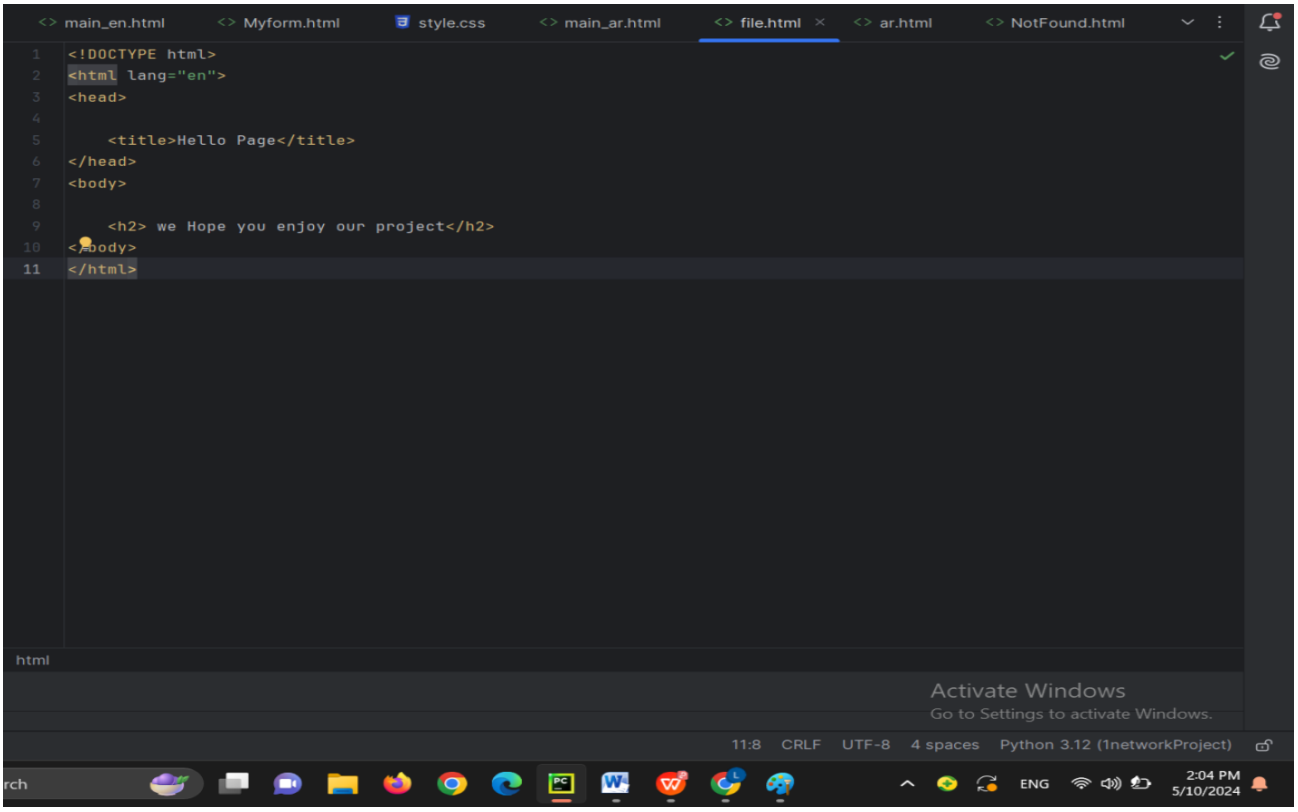
```
    <h2></h2>
    <!-- the element for hyper links   -->
    <a href="/Myform.html" target="_blank">
        <button id = "button1"> Myform</button>
    </a>
</h2>
<h2>
    <a href="https://www.w3schools.com/python/python_syntax.asp" target="_blank">
    <button id = "button2"> W3schools</button>
</a>
</h2>
</body>
</html>
```

html › body › div#intro-container

- HTML (notfound.html)

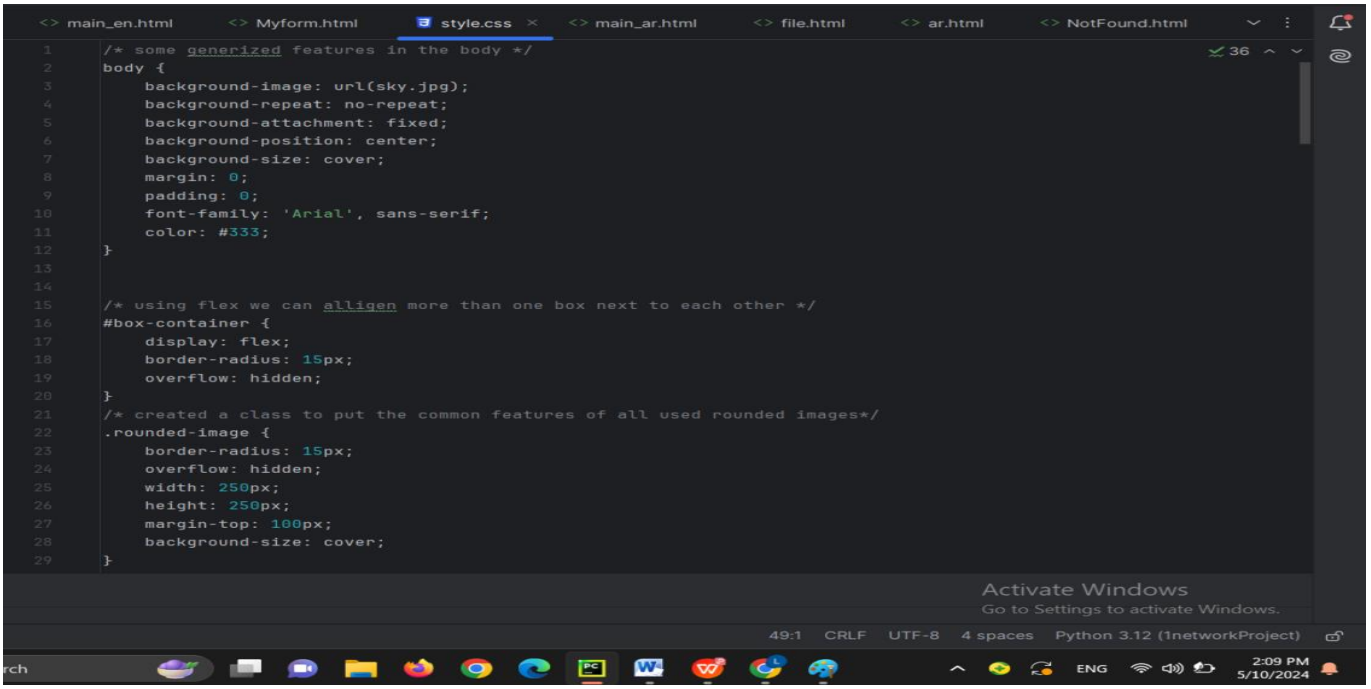


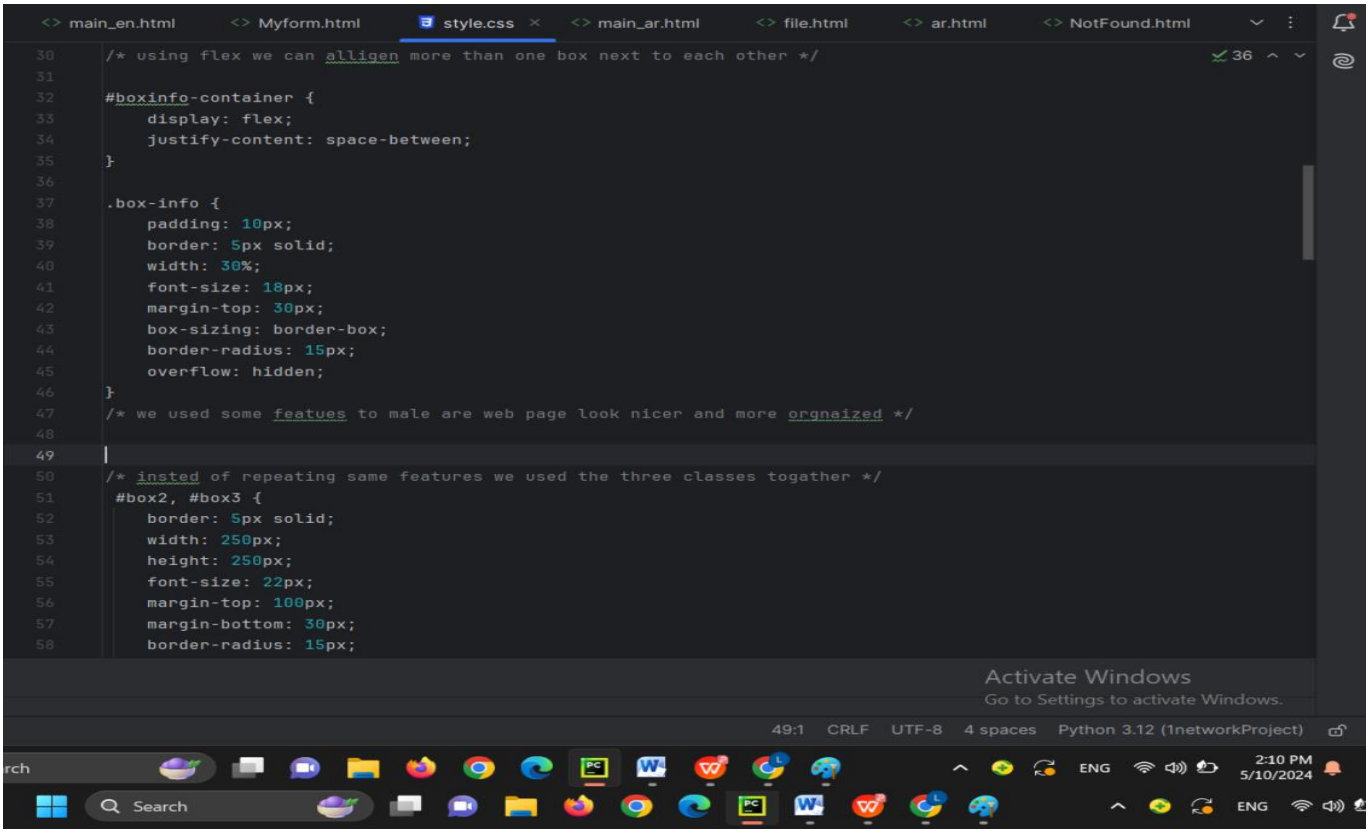
- HTML (file.html)

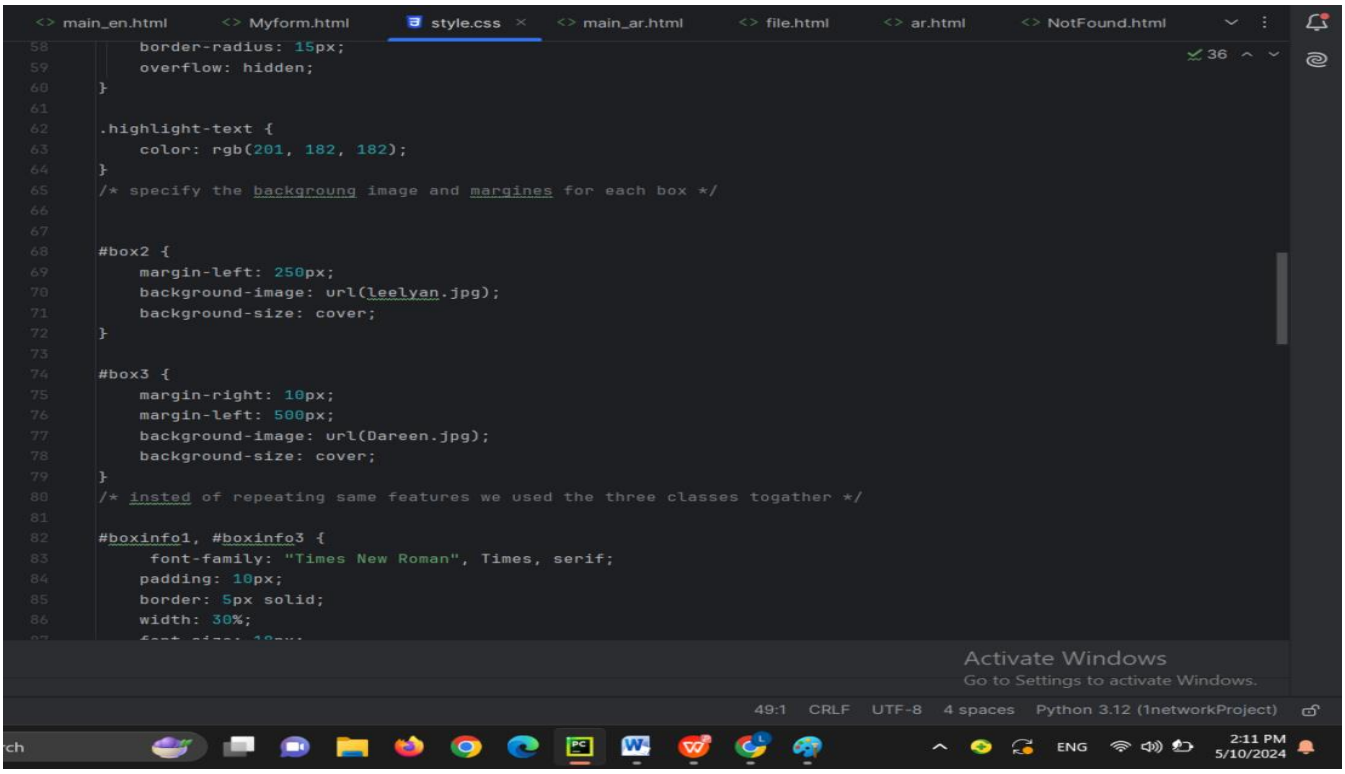


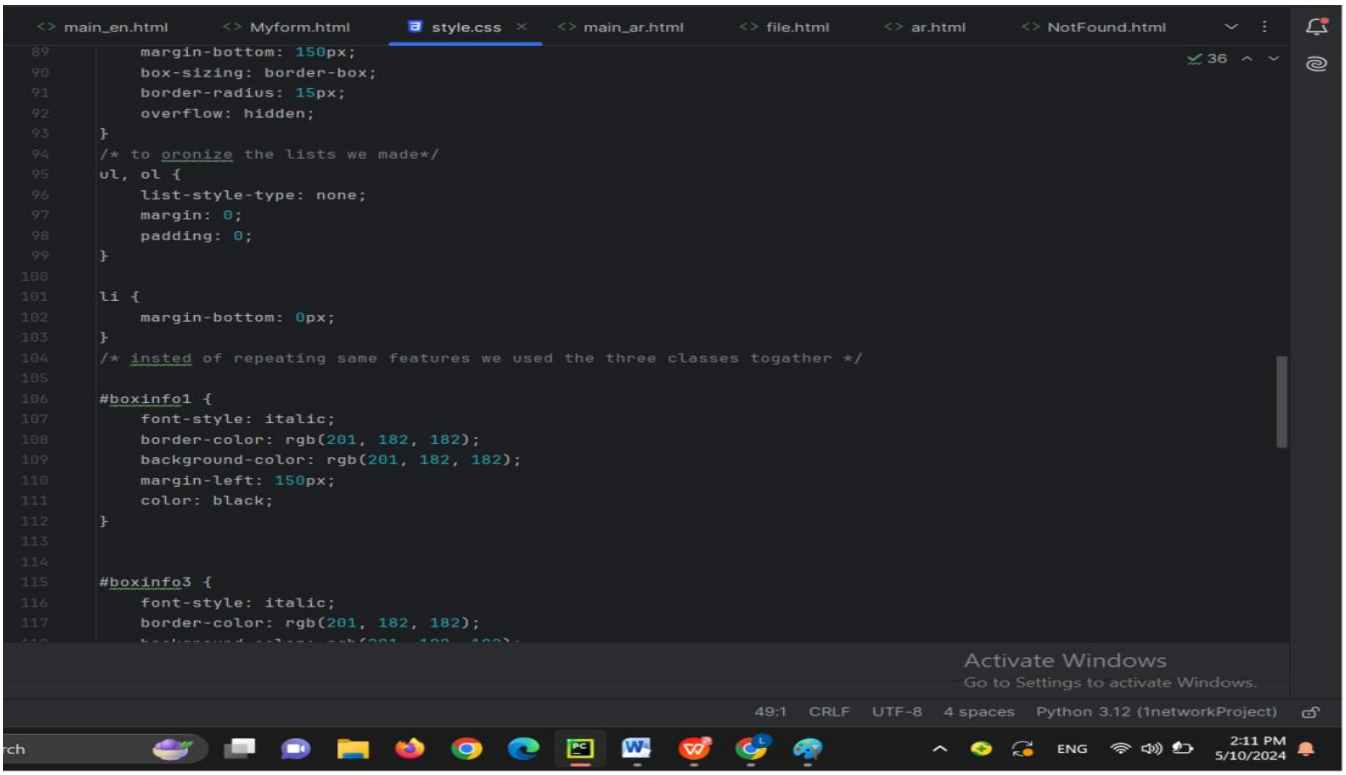
- style.css

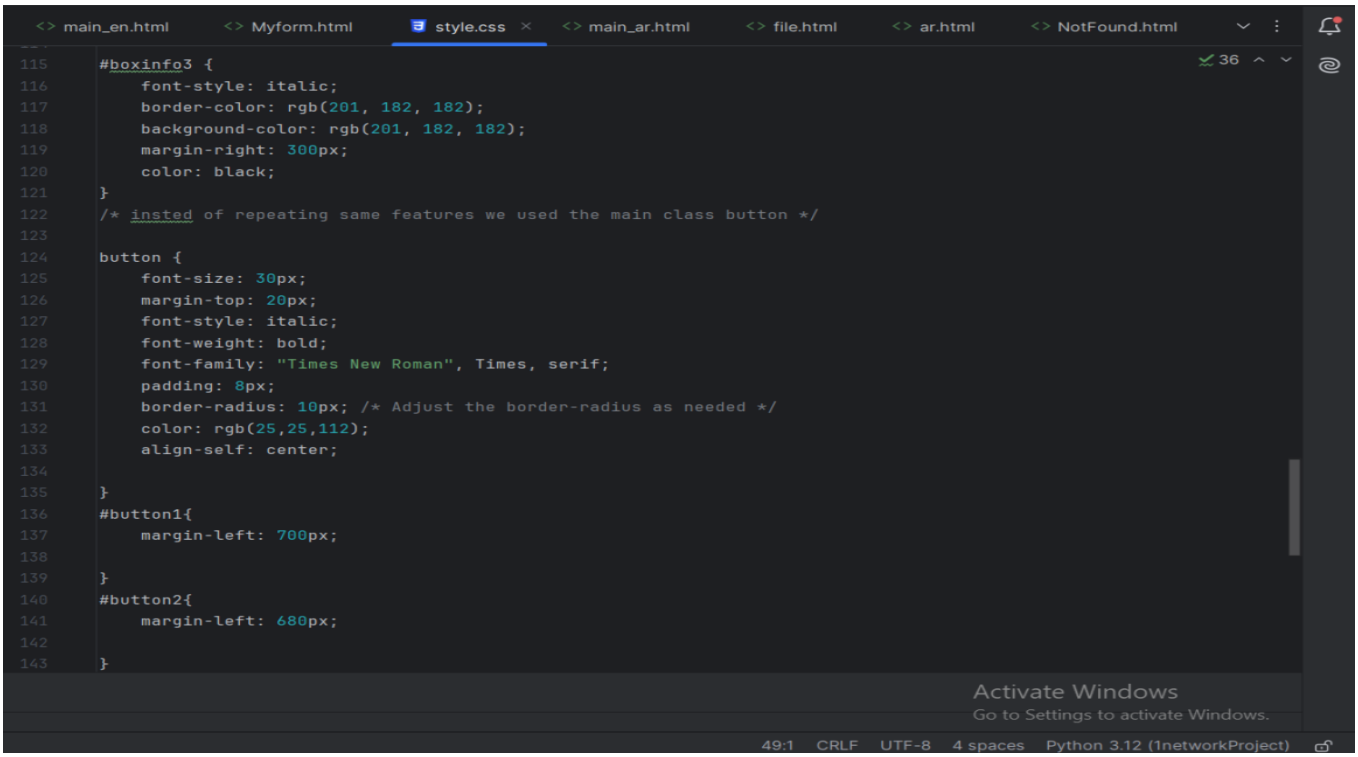
Figure 26:cssenglish version

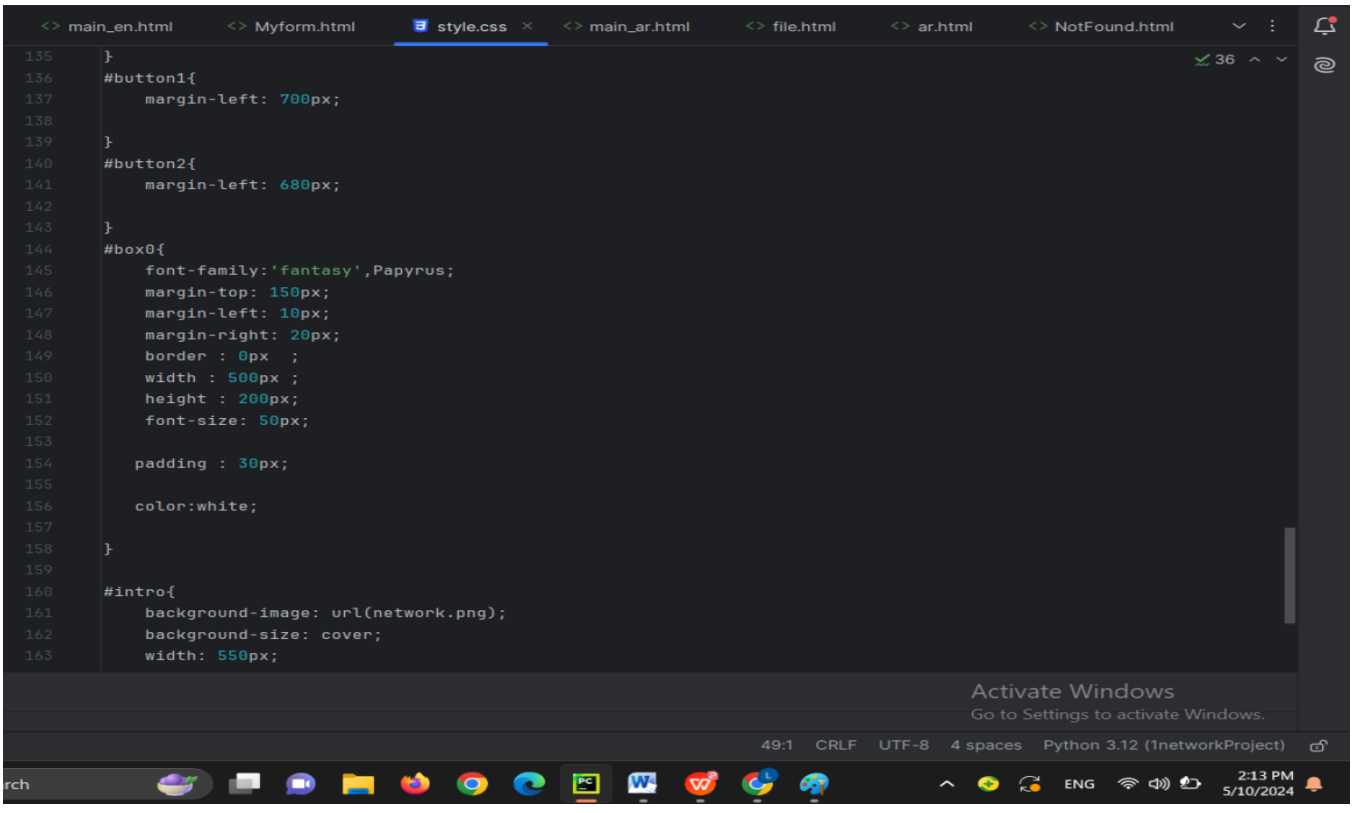


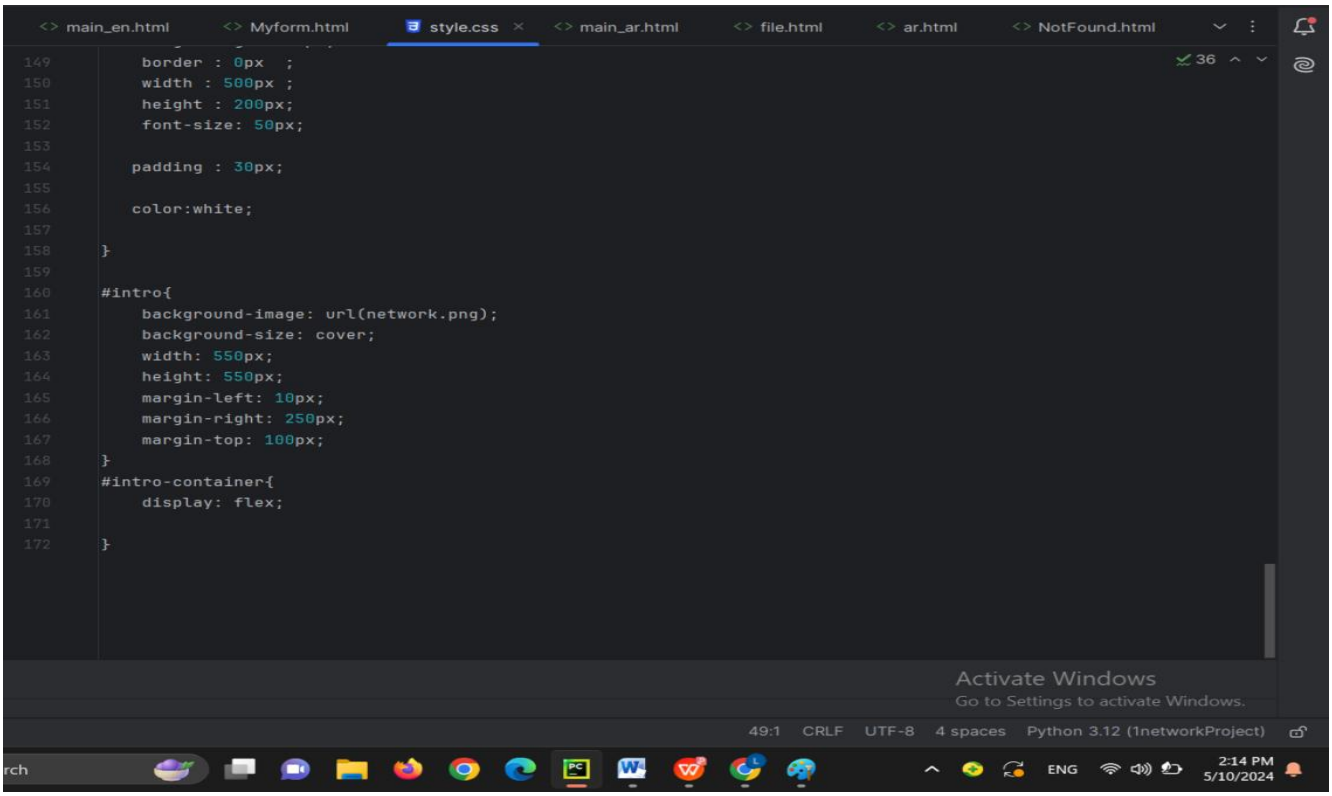



main_en.html
Myform.html
style.css
main_ar.html
file.html
ar.html
NotFound.html
border-radius: 15px;
overflow: hidden;
}
.highlight-text {
color: rgb(201, 182, 182);
}
/* specify the backgroung image and margines for each box */
#box2 {
margin-left: 250px;
background-image: url(leelyan.jpg);
background-size: cover;
}
#box3 {
margin-right: 10px;
margin-left: 500px;
background-image: url(Dareen.jpg);
background-size: cover;
}
/* insted of repeating same features we used the three classes togather */
#boxinfo1, #boxinfo3 {
font-family: "Times New Roman", Times, serif;
padding: 10px;
border: 5px solid;
width: 30%;
Activate Windows
Go to Settings to activate Windows.
49:1 CRLF UTF-8 4 spaces Python 3.12 (1networkProject)
ENG
2:11 PM
5/10/2024


main_en.html
Myform.html
style.css
main_ar.html
file.html
ar.html
NotFound.html
margin-bottom: 150px;
box-sizing: border-box;
border-radius: 15px;
overflow: hidden;
}
/* to oronize the lists we made*/
ul, ol {
list-style-type: none;
margin: 0;
padding: 0;
}
li {
margin-bottom: 0px;
}
/* insted of repeating same features we used the three classes togather */
#boxinfo1 {
font-style: italic;
border-color: rgb(201, 182, 182);
background-color: rgb(201, 182, 182);
margin-left: 150px;
color: black;
}
#boxinfo3 {
font-style: italic;
border-color: rgb(201, 182, 182);
Activate Windows
Go to Settings to activate Windows.
49:1 CRLF UTF-8 4 spaces Python 3.12 (1networkProject)
ENG
2:11 PM
5/10/2024

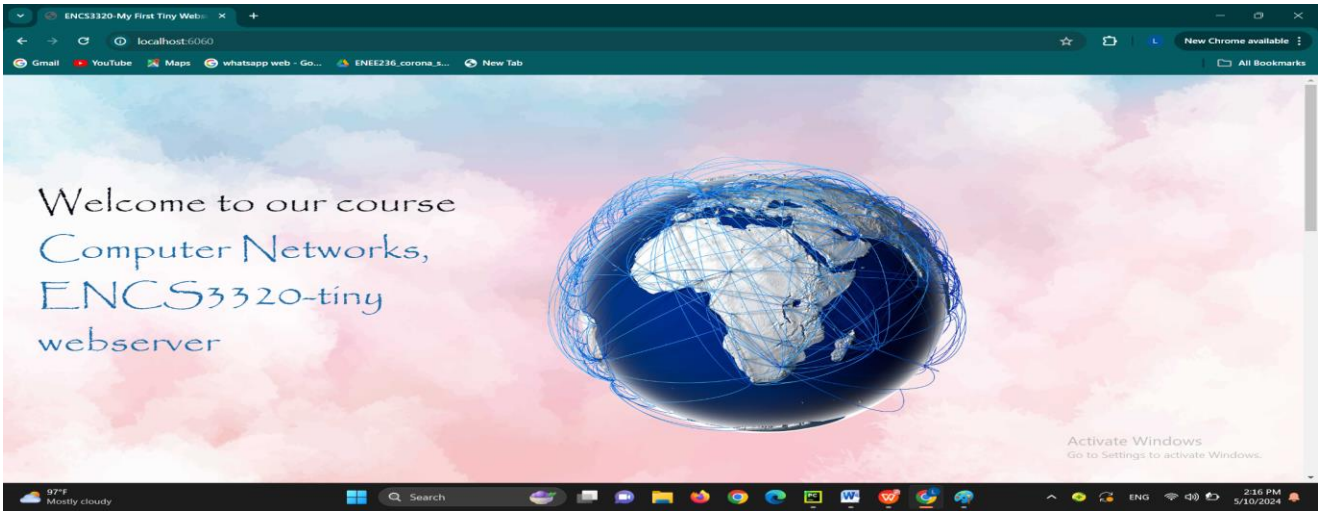

main_en.html
Myform.html
style.css
main_ar.html
file.html
ar.html
NotFound.html
#boxinfo3 {
font-style: italic;
border-color: rgb(201, 182, 182);
background-color: rgb(201, 182, 182);
margin-right: 300px;
color: black;
}
/* insted of repeating same features we used the main class button */
button {
font-size: 30px;
margin-top: 20px;
font-style: italic;
font-weight: bold;
font-family: "Times New Roman", Times, serif;
padding: 8px;
border-radius: 10px; /* Adjust the border-radius as needed */
color: rgb(25,25,112);
align-self: center;
}
#button1{
margin-left: 700px;
}
#button2{
margin-left: 680px;
}
Activate Windows
Go to Settings to activate Windows.
49:1 CRLF UTF-8 4 spaces Python 3.12 (1networkProject)

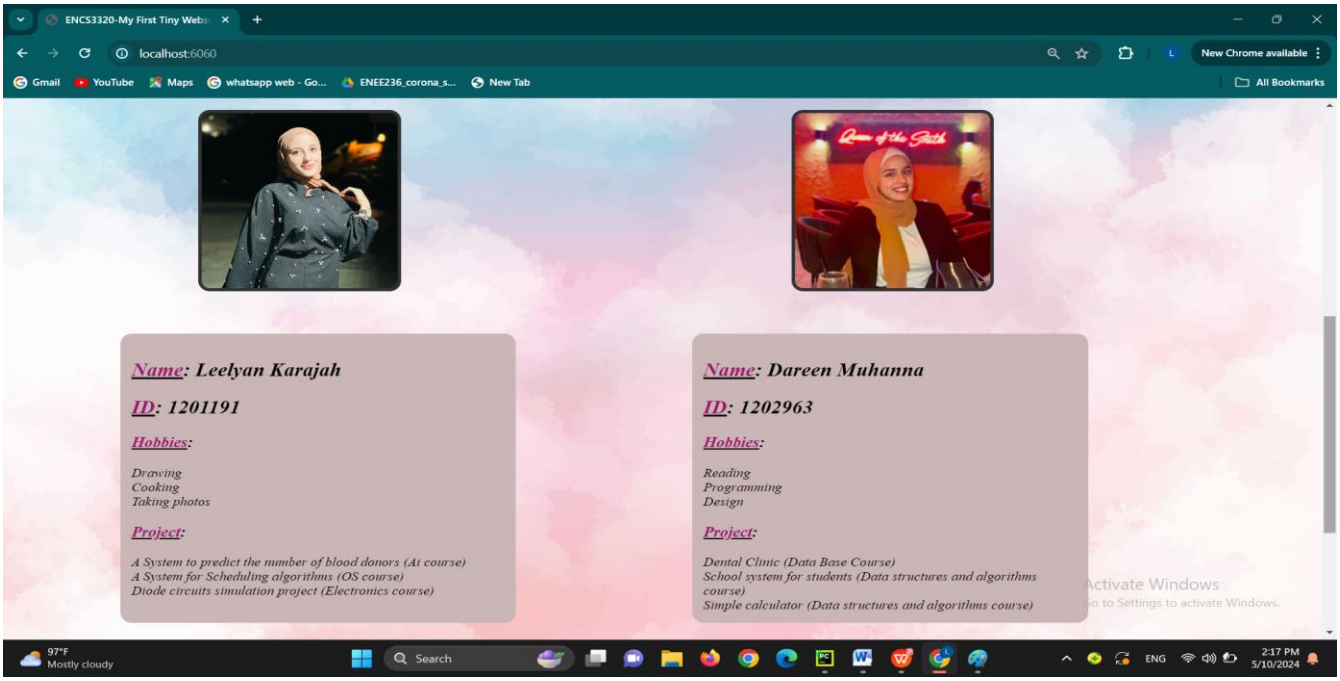

main_en.html
Myform.html
style.css
main_ar.html
file.html
ar.html
NotFound.html
#button1{
margin-left: 700px;
}
#button2{
margin-left: 680px;
}
#box0{
font-family:'fantasy',Papyrus;
margin-top: 150px;
margin-left: 10px;
margin-right: 20px;
border : 0px ;
width : 500px ;
height : 200px;
font-size: 50px;
padding : 30px;
color:white;
}
#intro{
background-image: url(network.png);
background-size: cover;
width: 550px;
Activate Windows
Go to Settings to activate Windows.
49:1 CRLF UTF-8 4 spaces Python 3.12 (1networkProject)
ENG
2:13 PM
5/10/2024

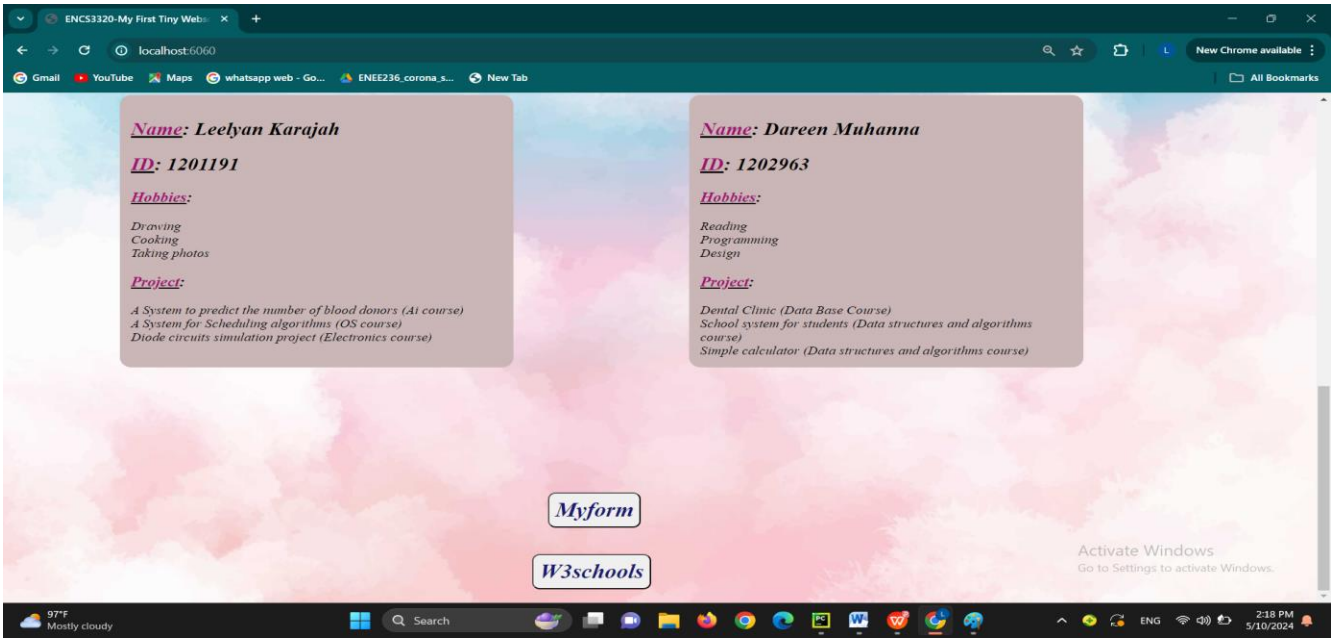


This CSS code aims to create a visually appealing and organized layout by using flexbox, rounded images, specific fonts, and carefully chosen colors.

- **Web server**

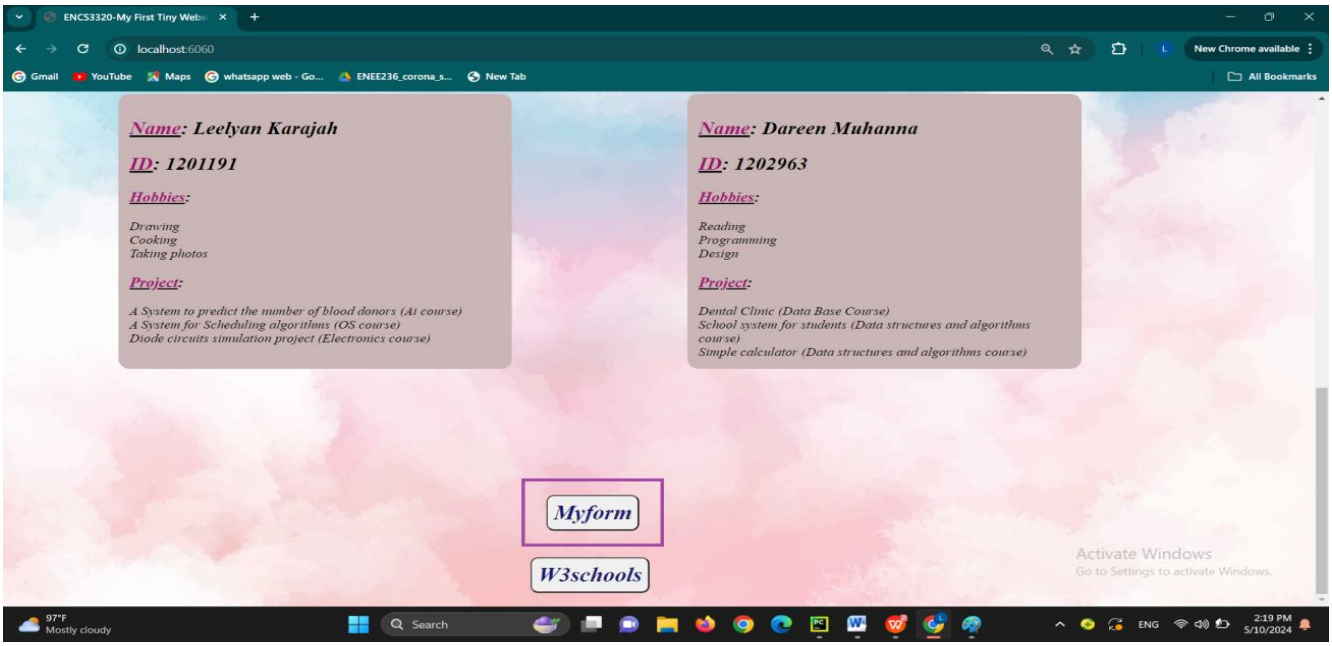




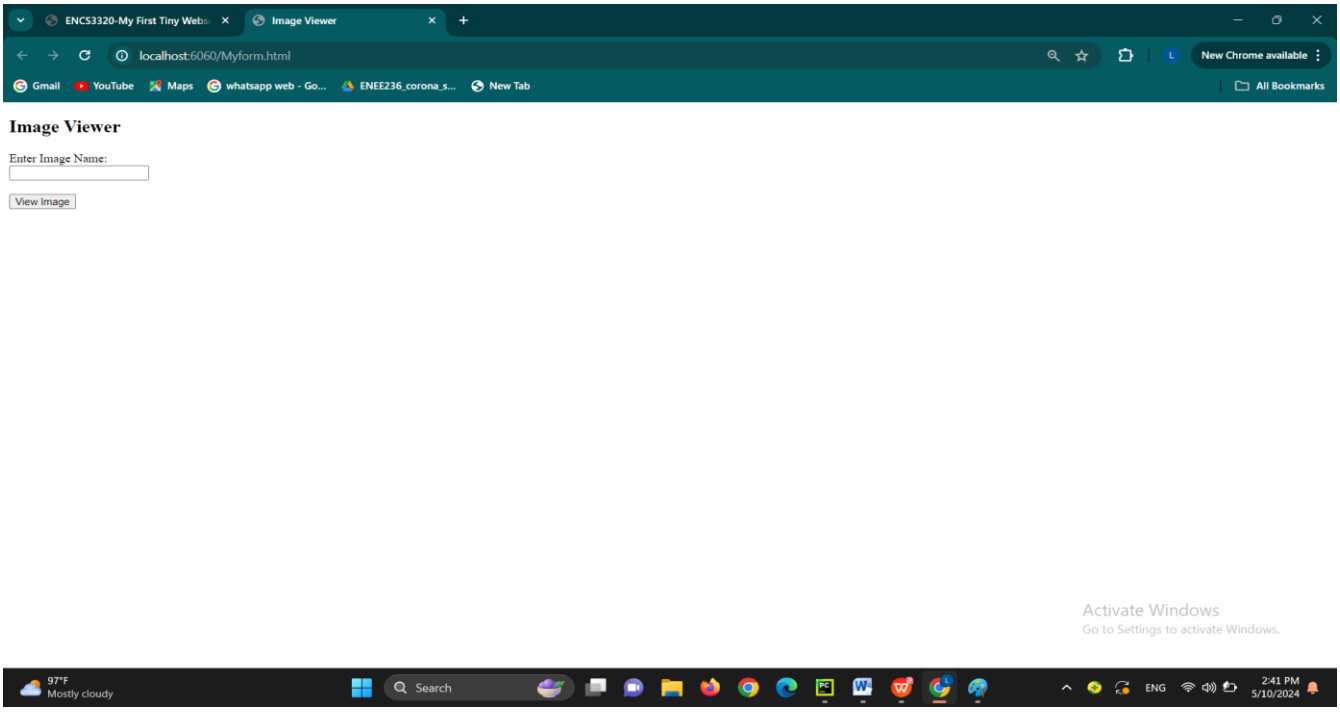


In the figures above we made sure to follow the instructions as given in the project description by setting the title of the page, the welcome phrase with the right colors, the group member names, IDs, Hobbies and preformed projects.

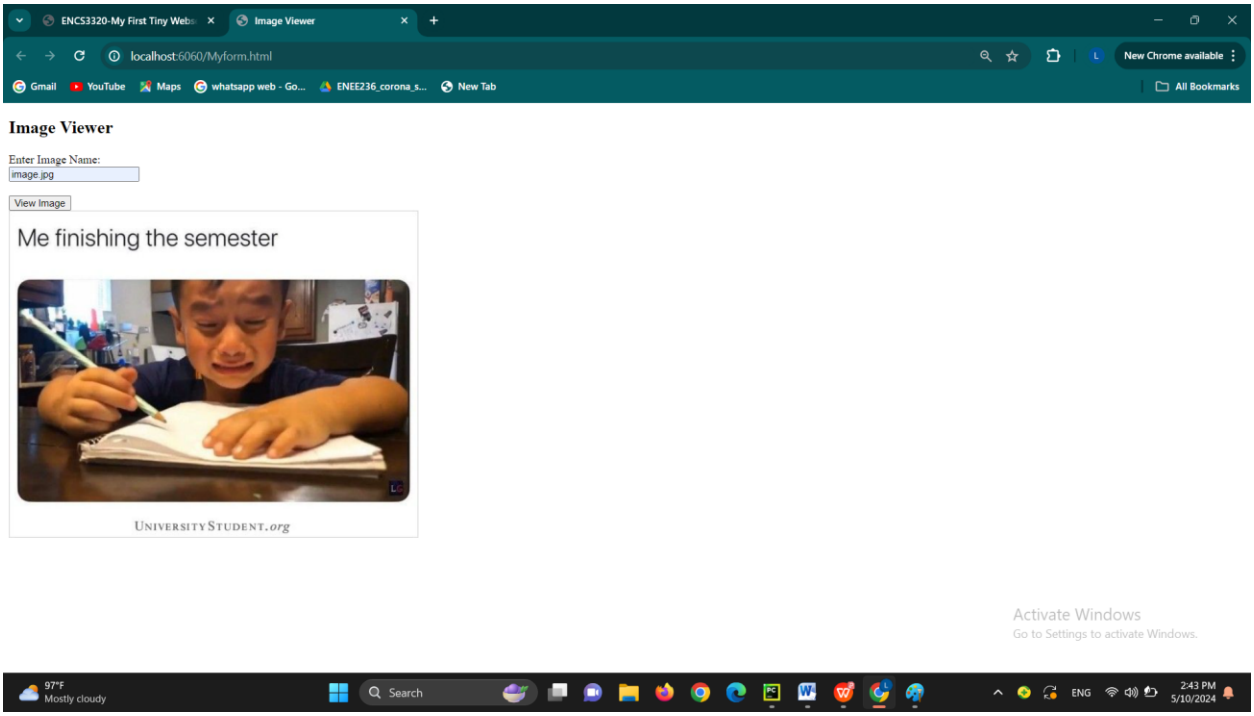
We also used CSS to enhance the frontend part of the project in every HTML file. **In the following figures, we will show the pages which our main page redirect you (by using hyper links).**

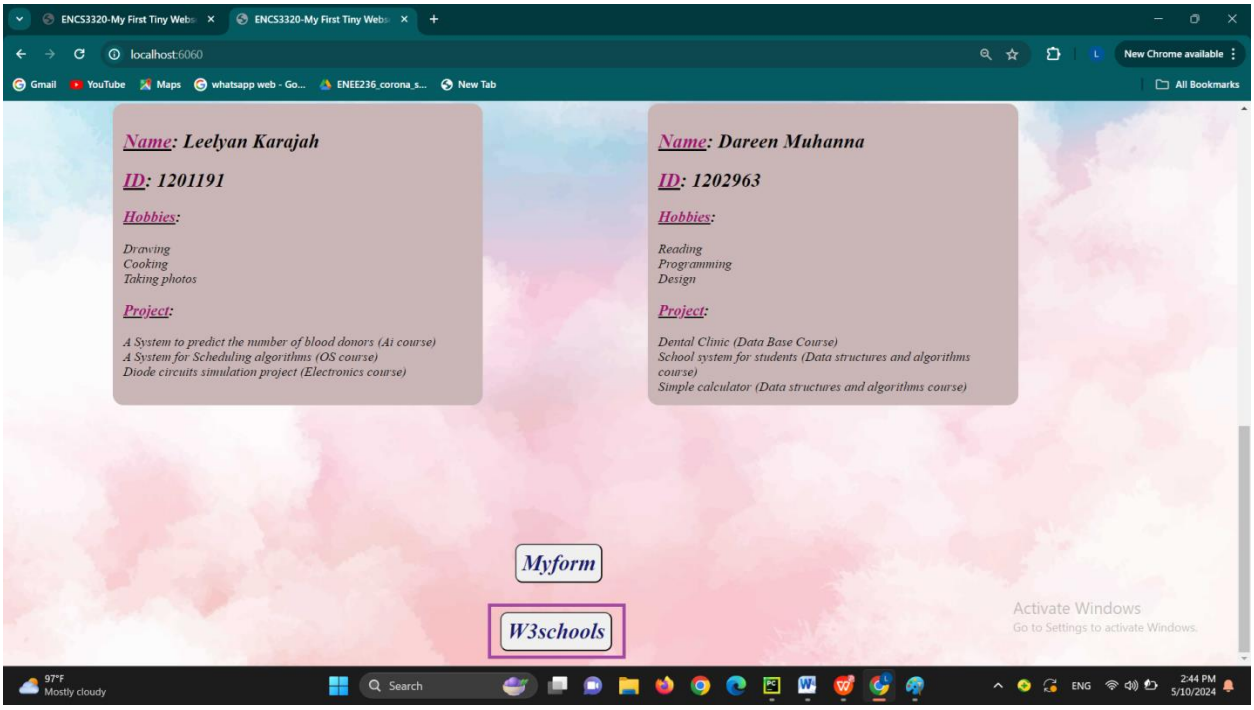


The My form button redirect you to the following html page (a local html page).

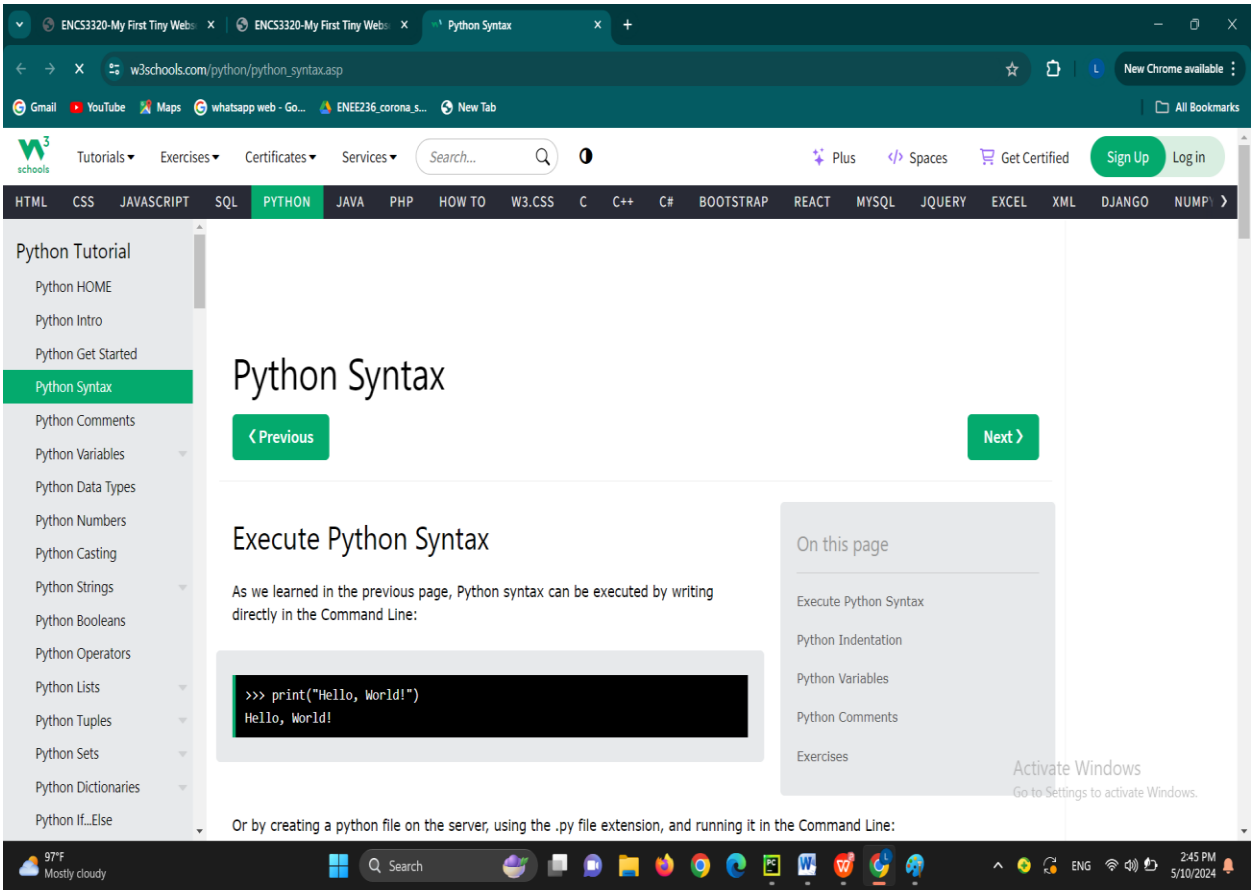


In the box you can but any image name ant it will appears .


ENCS3320-My First Tiny Webs
Image Viewer
localhost:6060/Myform.html
New Chrome available
Gmail
YouTube
Maps
whatsapp web - Go...
ENEE236_corona_s...
New Tab
All Bookmarks
Image Viewer
Enter Image Name:
image.jpg
View Image
Me finishing the semester
UNIVERSITY STUDENT.org
Activate Windows
Go to Settings to activate Windows.
97°F
Mostly cloudy
Search
ENG
2:43 PM
5/10/2024


ENCS3320-My First Tiny Webs
ENCS3320-My First Tiny Webs
localhost:6060
New Chrome available
Gmail
YouTube
Maps
whatsapp web - Go...
ENEE236_corona_s...
New Tab
All Bookmarks
Name: Leelyan Karajah
ID: 1201191
Hobbies:
Drawing
Cooking
Taking photos
Project:
A System to predict the number of blood donors (Ai course)
A System for Scheduling algorithms (OS course)
Diode circuits simulation project (Electronics course)
Name: Dareen Muhanna
ID: 1202963
Hobbies:
Reading
Programming
Design
Project:
Dental Clinic (Data Base Course)
School system for students (Data structures and algorithms course)
Simple calculator (Data structures and algorithms course)
Myform
W3schools
Activate Windows
Go to Settings to activate Windows.
97°F
Mostly cloudy
Search
ENG
2:44 PM
5/10/2024

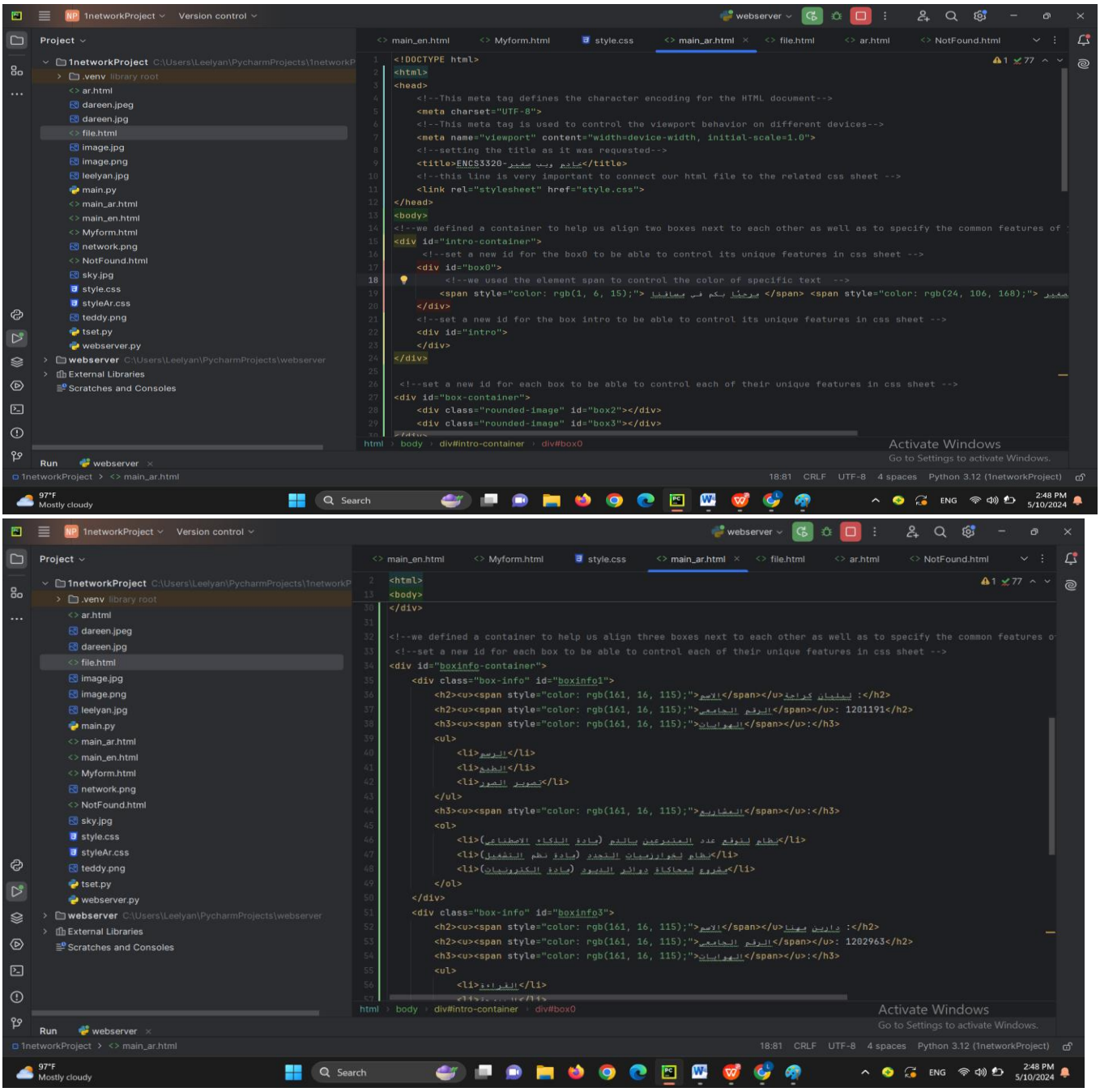
The W3schools button redirect you to the following page (a link that was provided by the project).

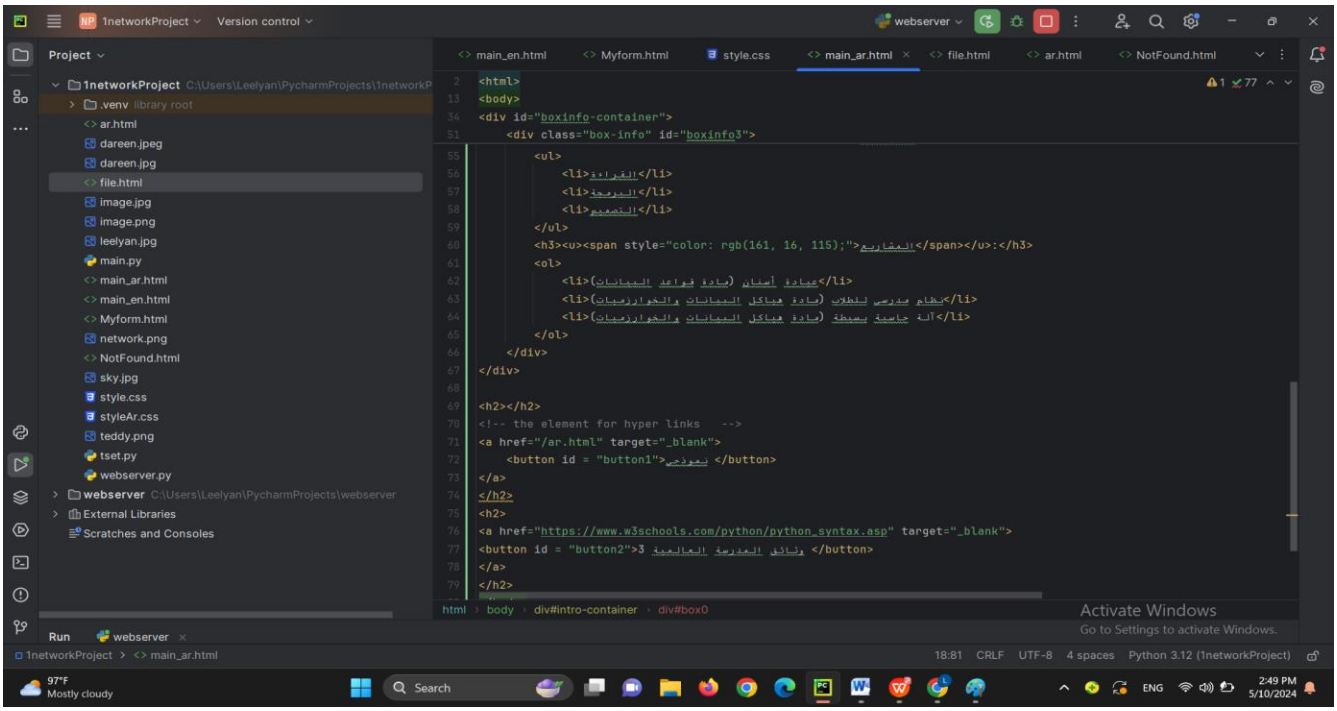


Arabic version

- HTML (main_ar.html)

Figure 27:HTMLARABIC version




1networkProject
Version control
webserver
Project
main_en.html
Myform.html
style.css
main_ar.html
file.html
ar.html
NotFound.html
1networkProject C:\Users\Leelyan\PycharmProjects\1networkP
.venv library root
ar.html
dareen.jpeg
dareen.jpg
file.html
image.jpg
image.png
leelyan.jpg
main.py
main_ar.html
main_en.html
Myform.html
network.png
NotFound.html
sky.jpg
style.css
styleAr.css
teddy.png
tset.py
webserver.py
webserver C:\Users\Leelyan\PycharmProjects\webserver
External Libraries
Scratches and Consoles
<html>
<body>

القراءة
البرمجة
التصميم

<h3><u>المشاريع</u>:</h3>

(مادة قواعد البيانات) عيادة أسنان
(مادة هياكل البيانات والخوارزميات) نظام مدرسي للطلاب
(مادة هياكل البيانات والخوارزميات) آلة حاسبة بسيطة

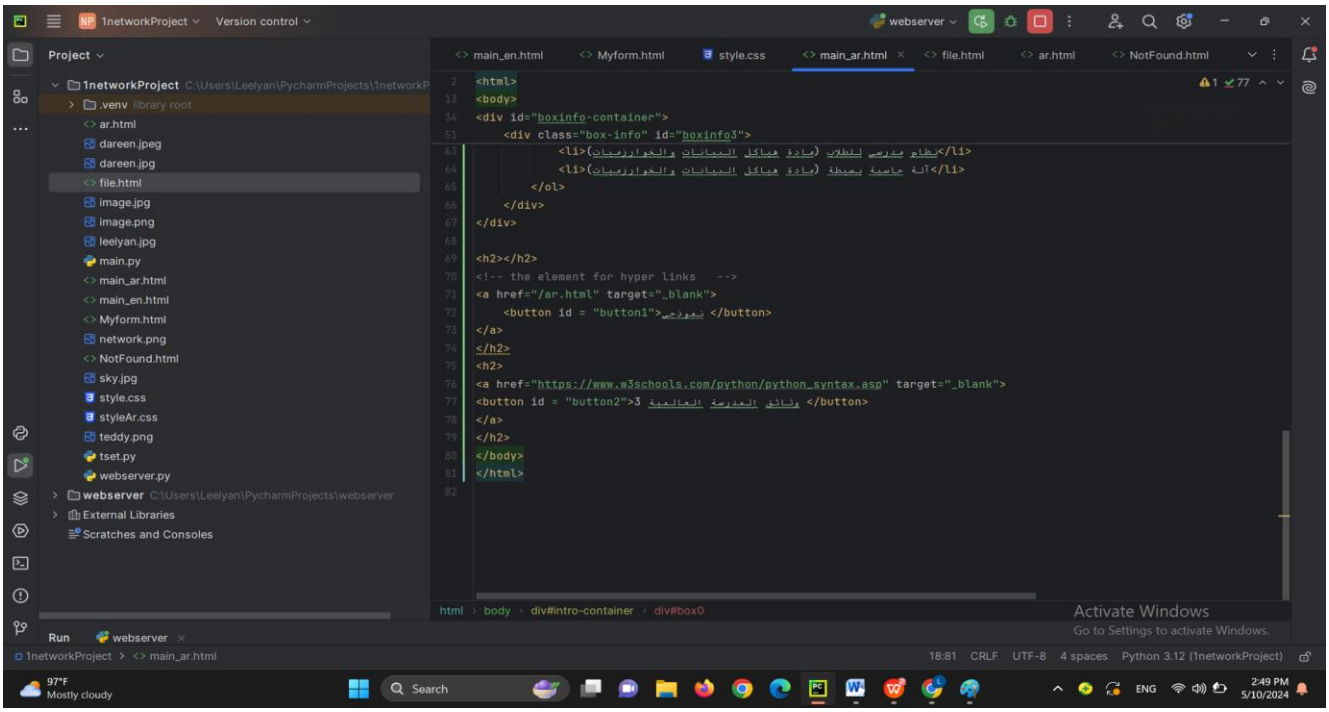
<h2></h2>
<!-- the element for hyper links -->

<button id = "button1">نموذجي </button>

</h2>
<h2>

<button id = "button2">3 وثائق المدرسة العالمية </button>

</h2>
html body div#intro-container div#box0
Activate Windows
Go to Settings to activate Windows.
Run
webserver
1networkProject main_ar.html
18:81 CRLF UTF-8 4 spaces Python 3.12 (1networkProject)
97°F
Mostly cloudy
Search
ENG
2:49 PM
5/10/2024


1networkProject
Version control
webserver
Project
main_en.html
Myform.html
style.css
main_ar.html
file.html
ar.html
NotFound.html
1networkProject C:\Users\Leelyan\PycharmProjects\1networkP
.venv library root
ar.html
dareen.jpeg
dareen.jpg
file.html
image.jpg
image.png
leelyan.jpg
main.py
main_ar.html
main_en.html
Myform.html
network.png
NotFound.html
sky.jpg
style.css
styleAr.css
teddy.png
tset.py
webserver.py
webserver C:\Users\Leelyan\PycharmProjects\webserver
External Libraries
Scratches and Consoles
(مادة هياكل البيانات والخوارزميات) نظام مدرسي للطلاب
(مادة هياكل البيانات والخوارزميات) آلة حاسبة بسيطة

<h2></h2>
<!-- the element for hyper links -->

<button id = "button1">نموذجي </button>

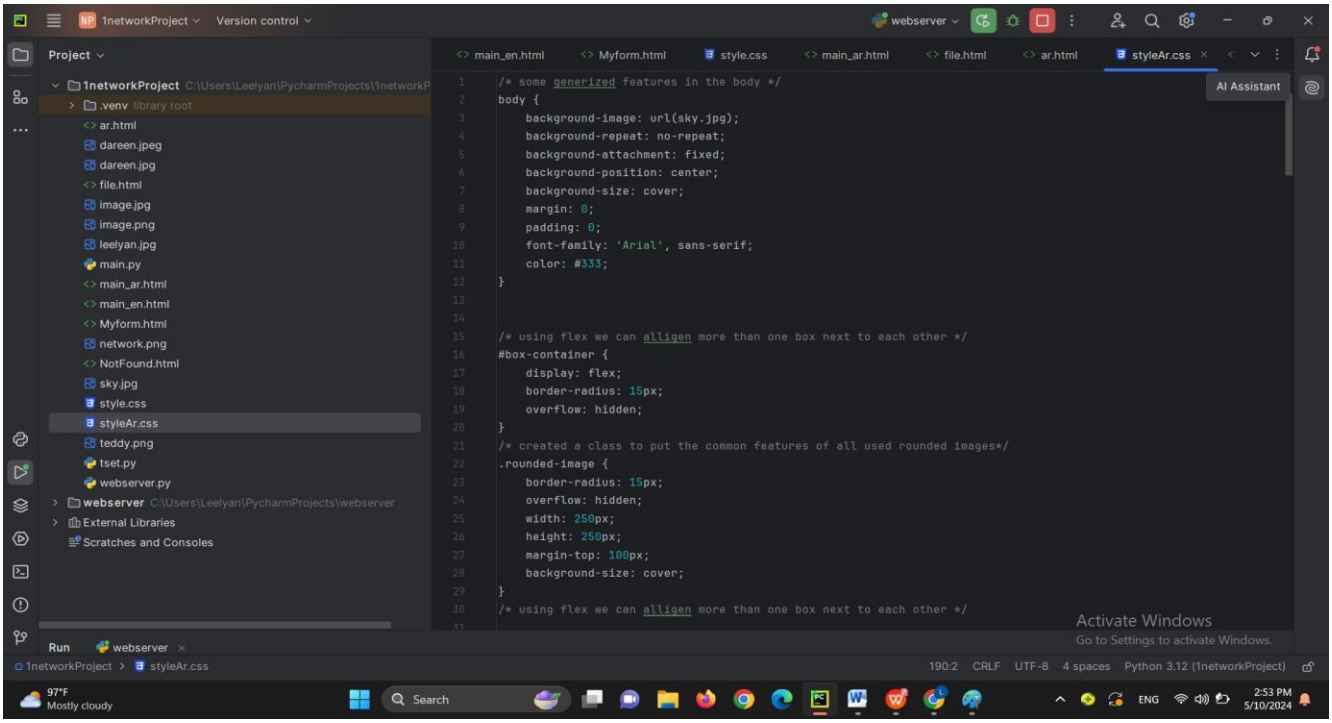
</h2>
<h2>

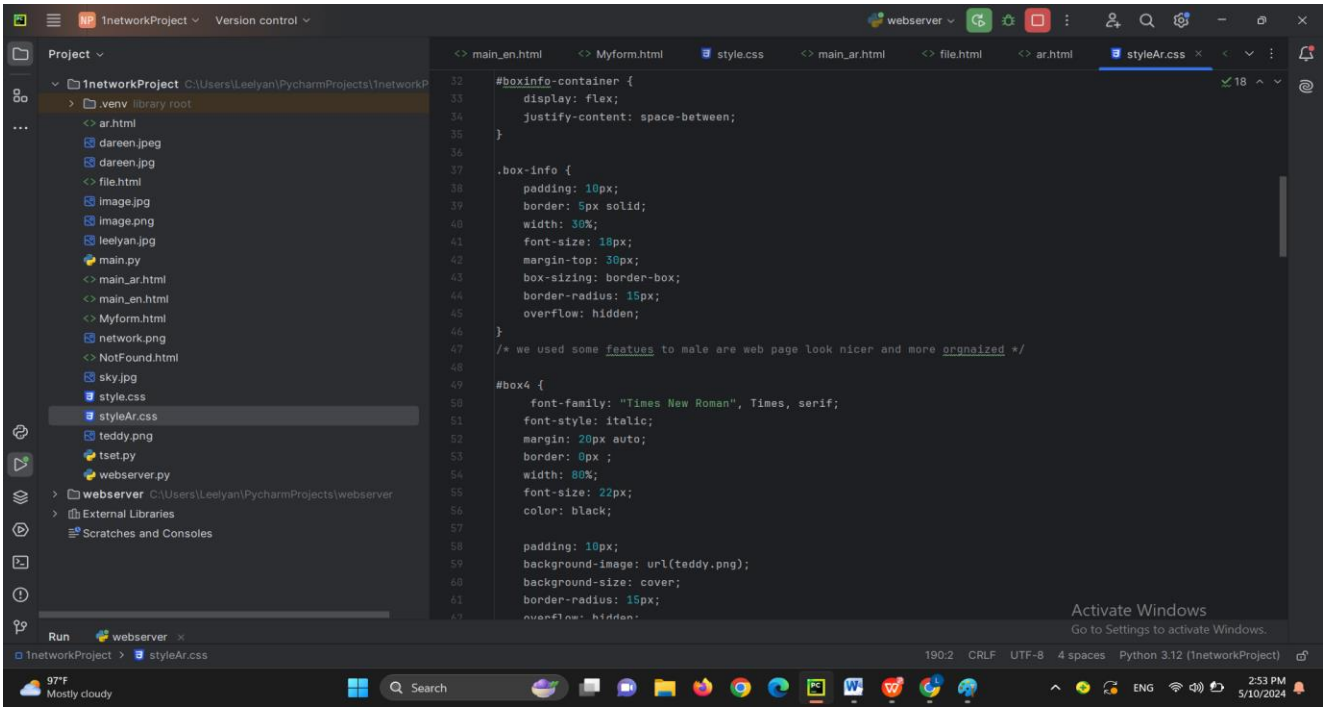
<button id = "button2">3 وثائق المدرسة العالمية </button>

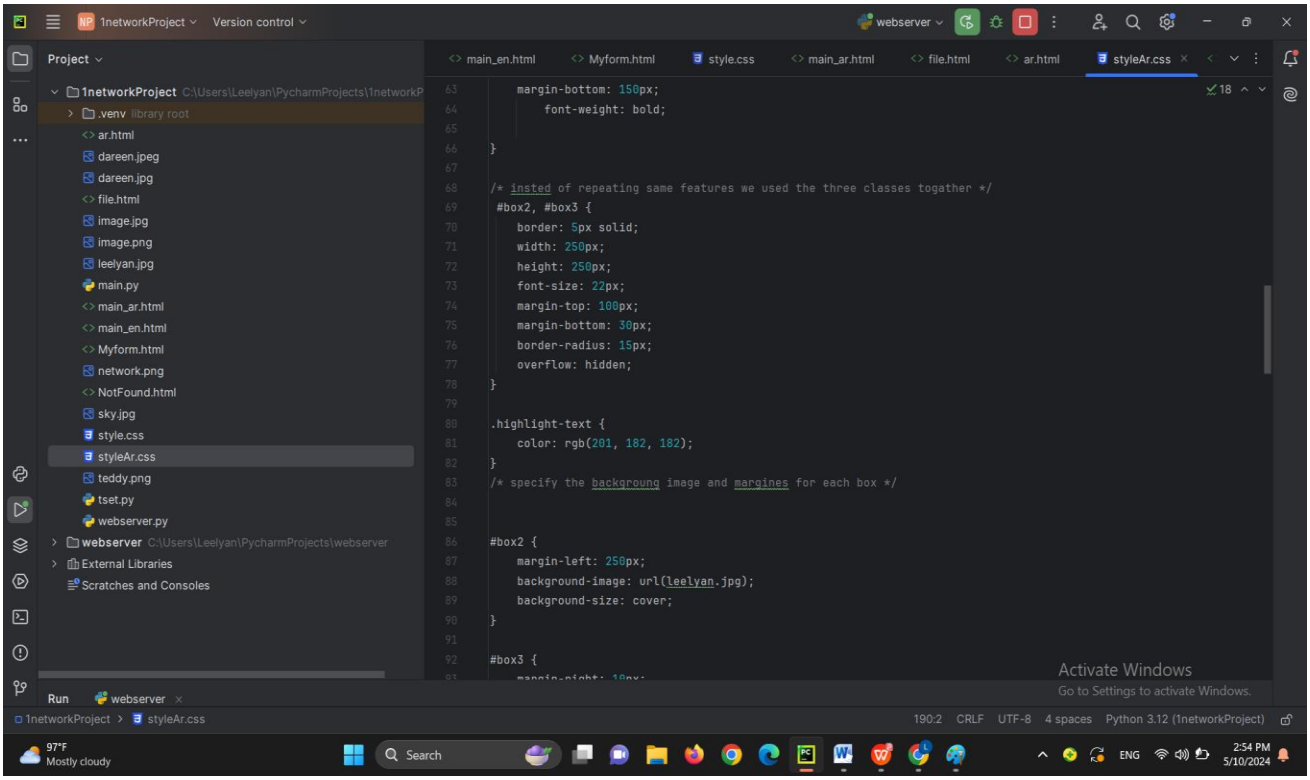
</h2>
</body>
</html>
html body div#intro-container div#box0
Activate Windows
Go to Settings to activate Windows.
Run
webserver
1networkProject main_ar.html
18:81 CRLF UTF-8 4 spaces Python 3.12 (1networkProject)
97°F
Mostly cloudy
Search
ENG
2:49 PM
5/10/2024

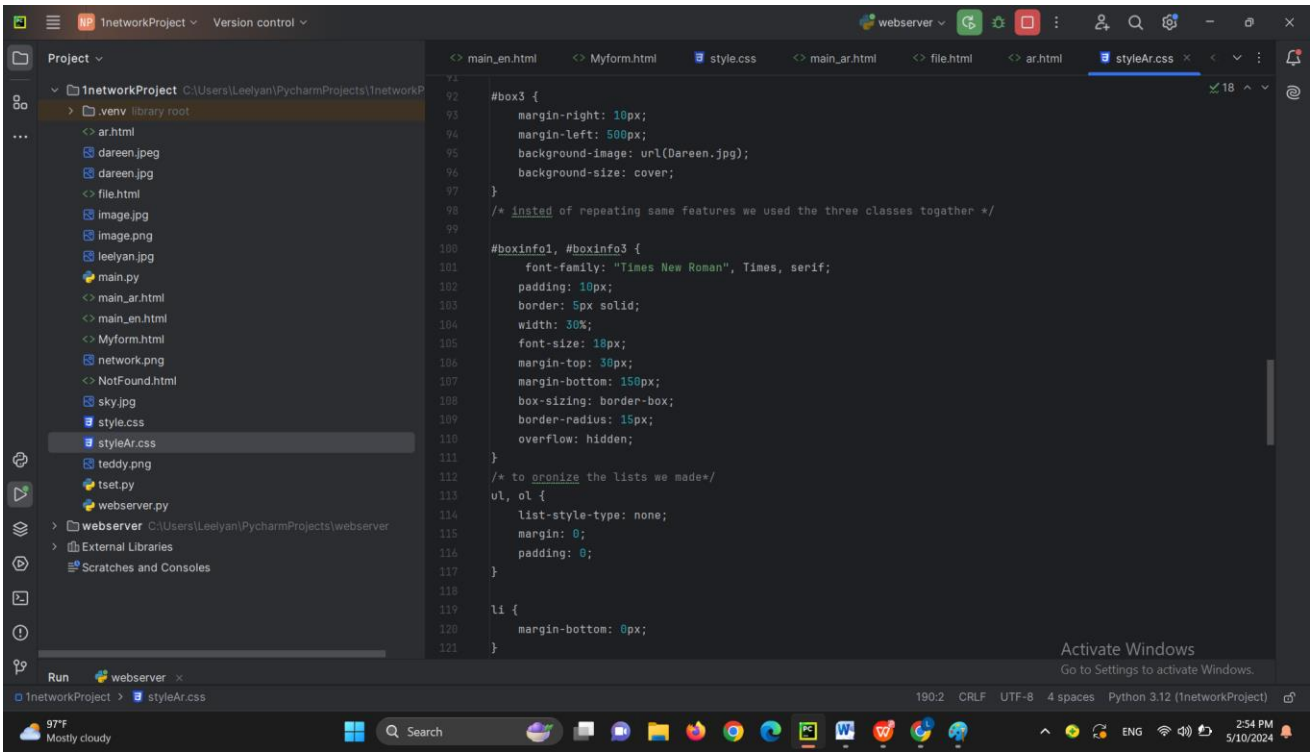
- styleAr.css

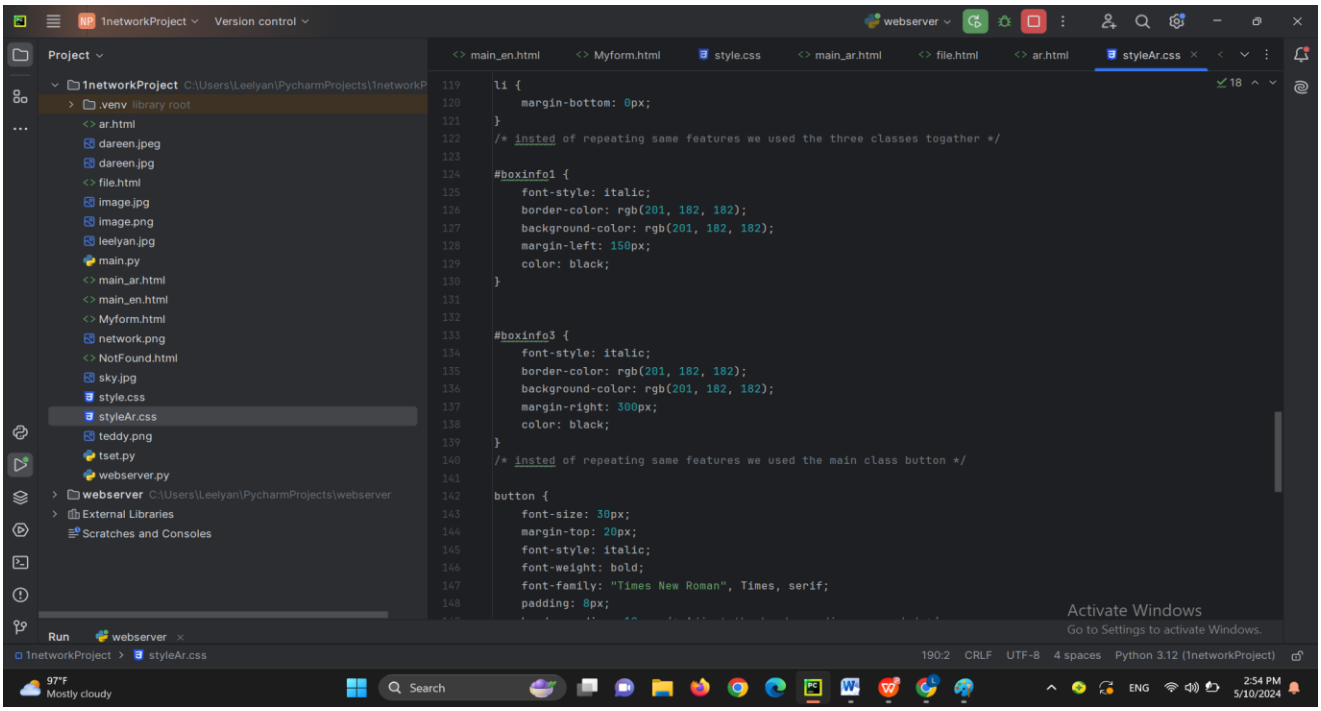
Figure 28:CSS arabic version

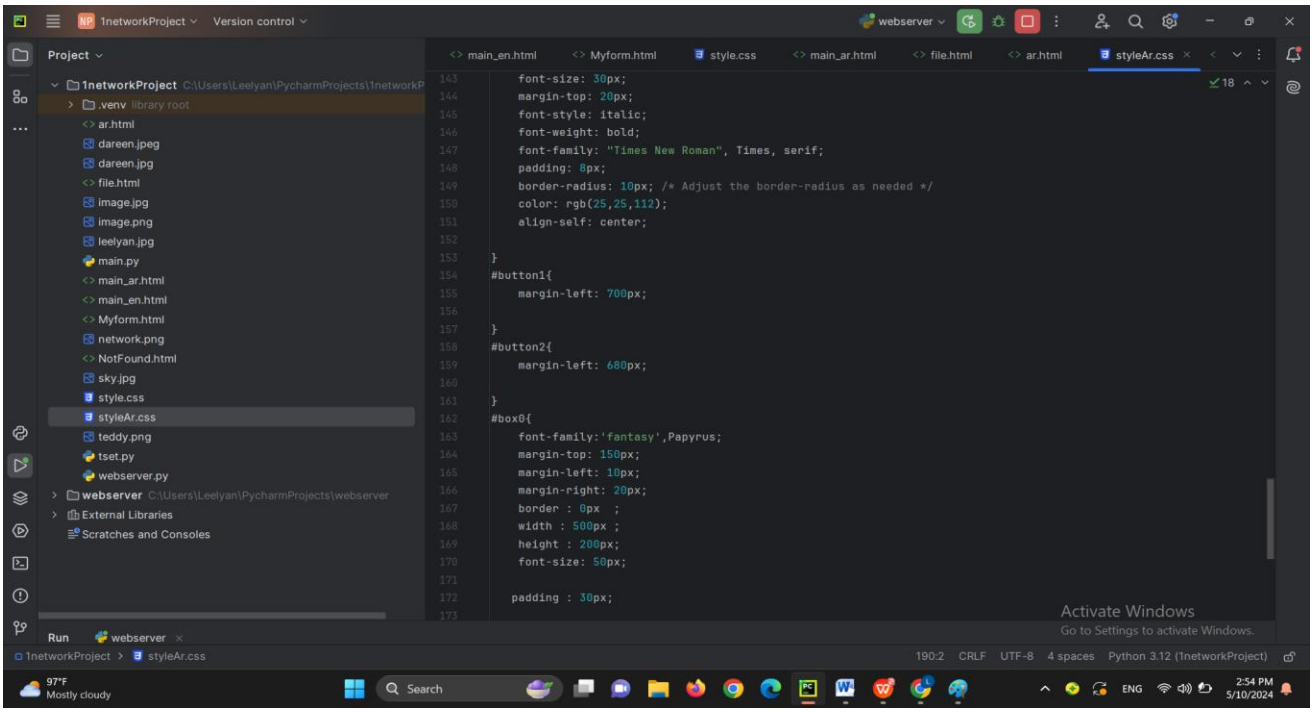


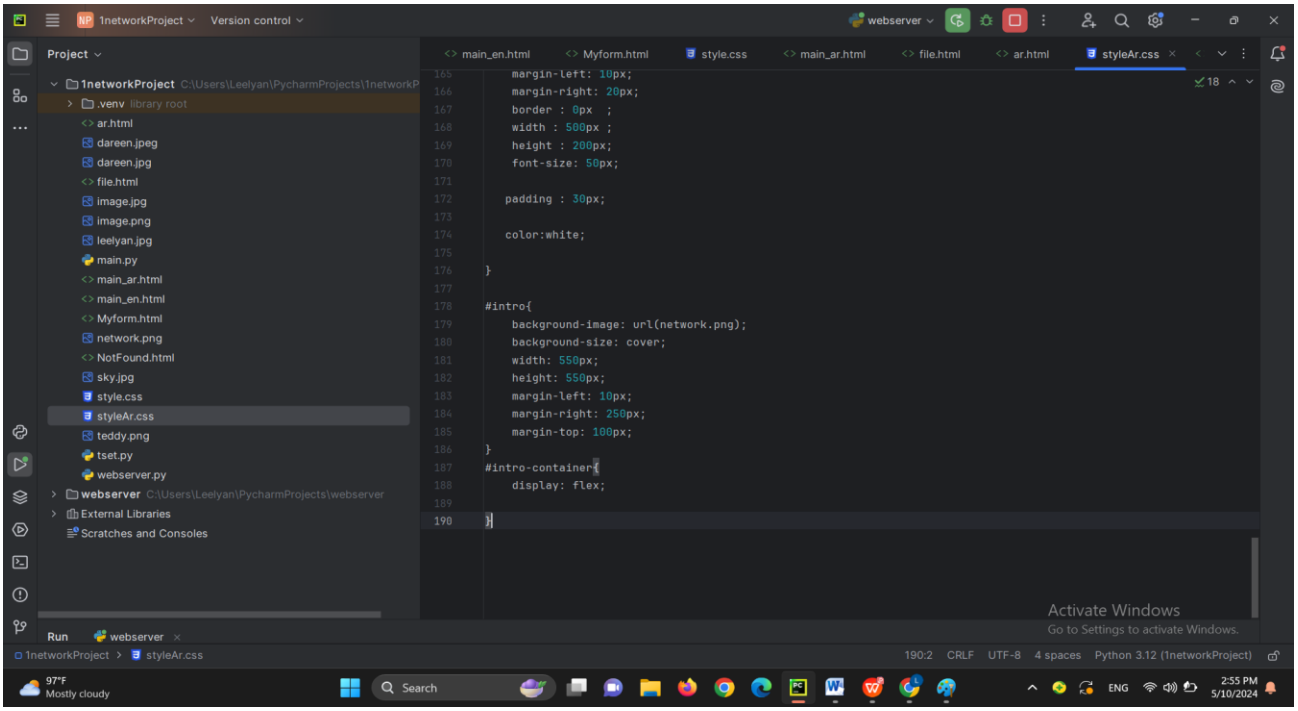



margin-bottom: 150px;
font-weight: bold;
}
/* insted of repeating same features we used the three classes togather */
#box2, #box3 {
border: 5px solid;
width: 250px;
height: 250px;
font-size: 22px;
margin-top: 100px;
margin-bottom: 30px;
border-radius: 15px;
overflow: hidden;
}
.highlight-text {
color: rgb(201, 182, 182);
}
/* specify the backgroung image and margines for each box */
#box2 {
margin-left: 250px;
background-image: url(leelyan.jpg);
background-size: cover;
}
#box3 {
Activate Windows
Go to Settings to activate Windows.

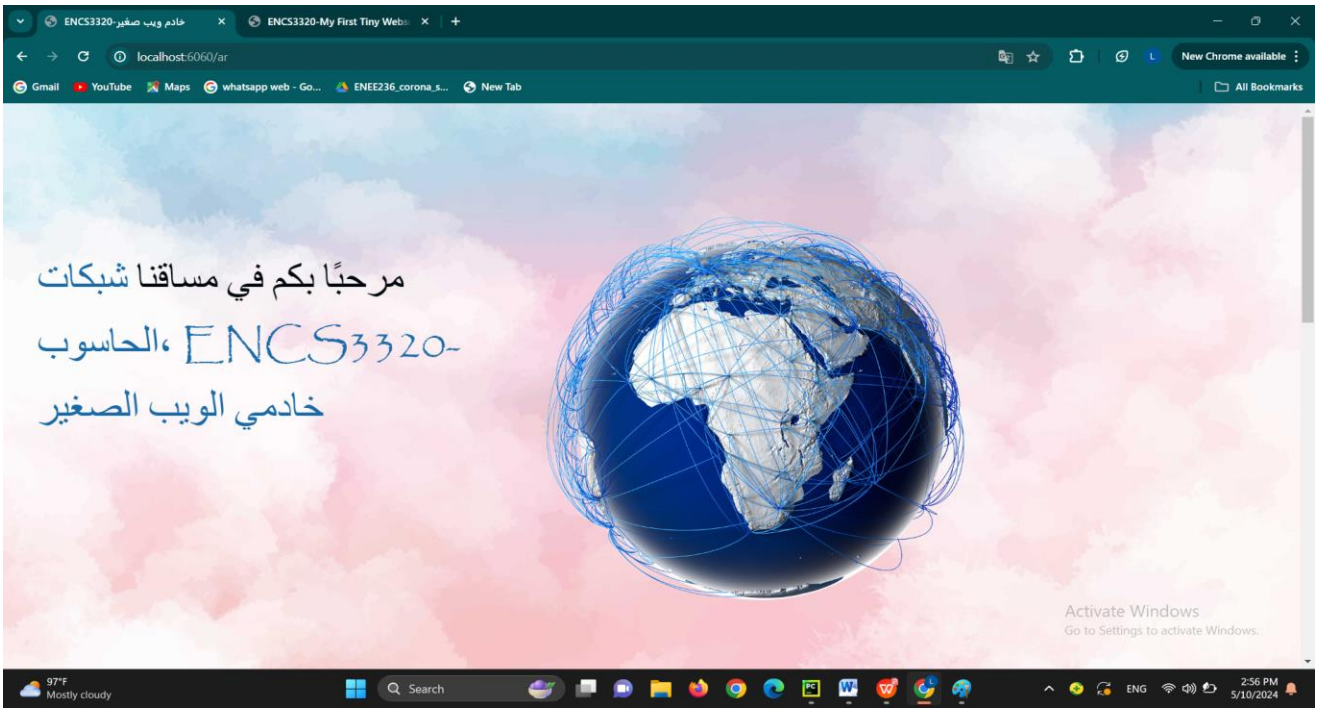

#box3 {
margin-right: 10px;
margin-left: 500px;
background-image: url(Dareen.jpg);
background-size: cover;
}
/* insted of repeating same features we used the three classes togather */
#boxinfo1, #boxinfo3 {
font-family: "Times New Roman", Times, serif;
padding: 10px;
border: 5px solid;
width: 30%;
font-size: 18px;
margin-top: 30px;
margin-bottom: 150px;
box-sizing: border-box;
border-radius: 15px;
overflow: hidden;
}
/* to oronize the lists we made*/
ul, ol {
list-style-type: none;
margin: 0;
padding: 0;
}
li {
margin-bottom: 0px;
}
Activate Windows
Go to Settings to activate Windows.

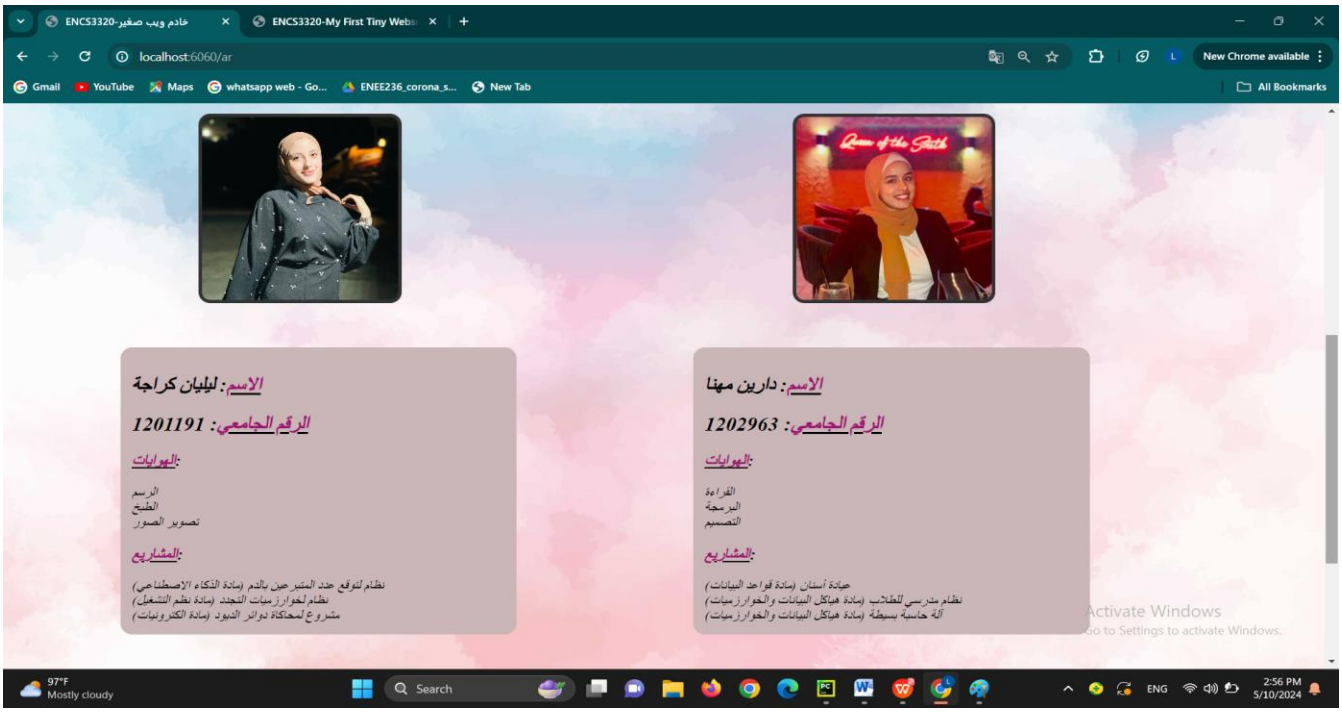

1networkProject
Version control
webserver
Project
main_en.html
Myform.html
style.css
main_ar.html
file.html
ar.html
styleAr.css
li {
margin-bottom: 0px;
}
/* insted of repeating same features we used the three classes togather */
#boxinfo1 {
font-style: italic;
border-color: rgb(201, 182, 182);
background-color: rgb(201, 182, 182);
margin-left: 150px;
color: black;
}
#boxinfo3 {
font-style: italic;
border-color: rgb(201, 182, 182);
background-color: rgb(201, 182, 182);
margin-right: 300px;
color: black;
}
/* insted of repeating same features we used the main class button */
button {
font-size: 30px;
margin-top: 20px;
font-style: italic;
font-weight: bold;
font-family: "Times New Roman", Times, serif;
padding: 8px;
Activate Windows
Go to Settings to activate Windows.
Run
webserver
190:2 CRLF UTF-8 4 spaces Python 3.12 (1networkProject)
97°F
Mostly cloudy
Search
ENG
2:54 PM
5/10/2024

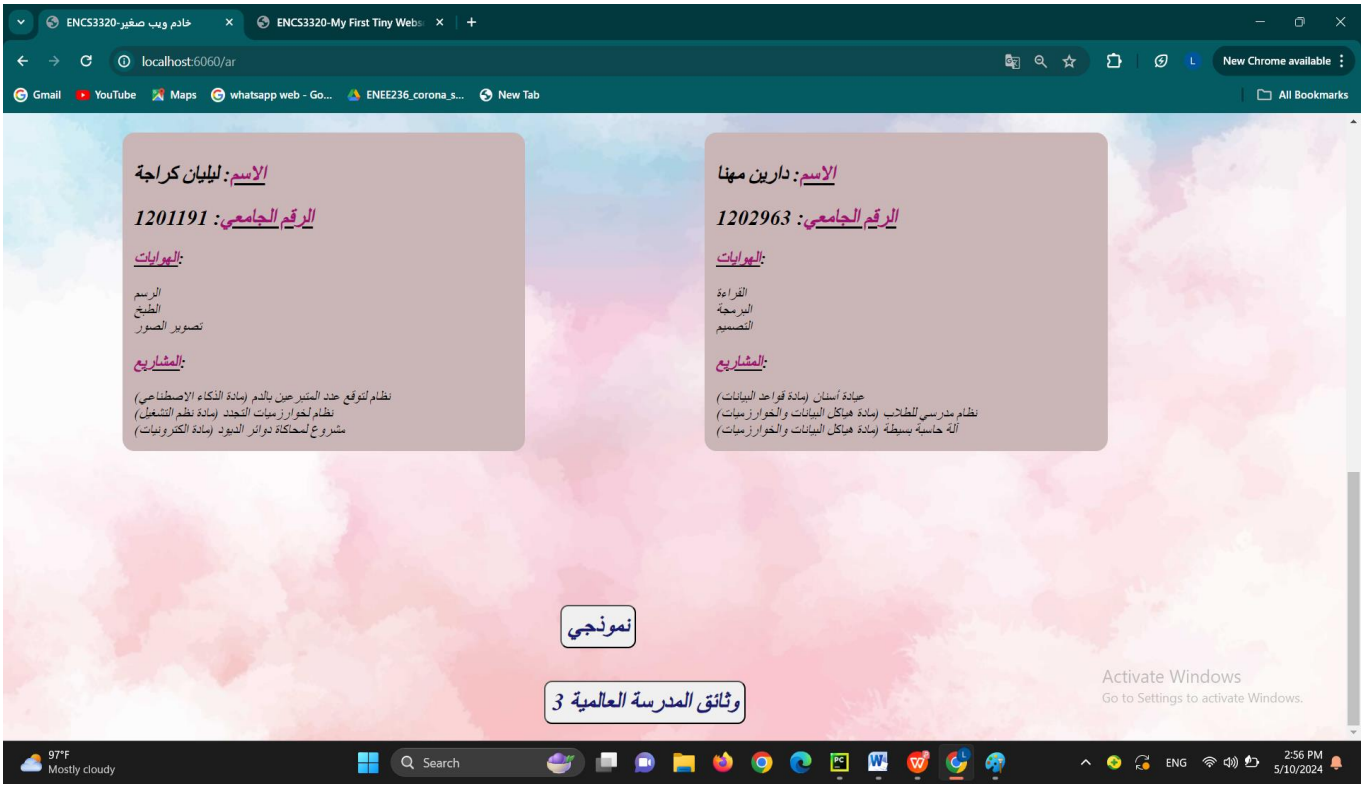

1networkProject
Version control
webserver
Project
main_en.html
Myform.html
style.css
main_ar.html
file.html
ar.html
styleAr.css
font-size: 30px;
margin-top: 20px;
font-style: italic;
font-weight: bold;
font-family: "Times New Roman", Times, serif;
padding: 8px;
border-radius: 10px; /* Adjust the border-radius as needed */
color: rgb(25,25,112);
align-self: center;
}
#button1{
margin-left: 700px;
}
#button2{
margin-left: 680px;
}
#box0{
font-family:'fantasy',Papyrus;
margin-top: 150px;
margin-left: 10px;
margin-right: 20px;
border : 0px ;
width : 500px ;
height : 200px;
font-size: 50px;
padding : 30px;
Activate Windows
Go to Settings to activate Windows.
Run
webserver
190:2 CRLF UTF-8 4 spaces Python 3.12 (1networkProject)
97°F
Mostly cloudy
Search
ENG
2:54 PM
5/10/2024

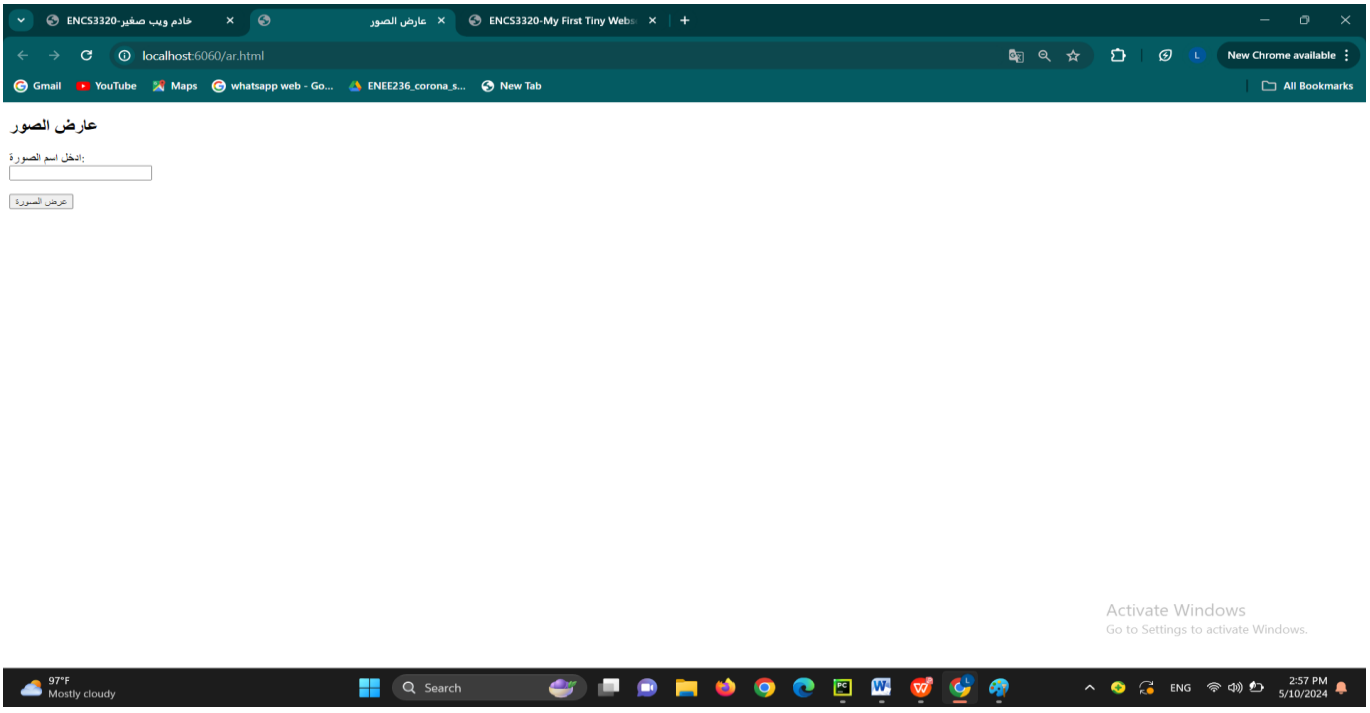

1networkProject
Version control
webserver
Project
1networkProject C:\Users\Leelyan\PycharmProjects\1networkP
.venv library root
ar.html
dareen.jpeg
dareen.jpg
file.html
image.jpg
image.png
leelyan.jpg
main.py
main_ar.html
main_en.html
Myform.html
network.png
NotFound.html
sky.jpg
style.css
styleAr.css
teddy.png
tset.py
webserver.py
webserver C:\Users\Leelyan\PycharmProjects\webserver
External Libraries
Scratches and Consoles
main_en.html
Myform.html
style.css
main_ar.html
file.html
ar.html
styleAr.css
margin-left: 10px;
margin-right: 20px;
border : 0px ;
width : 500px ;
height : 200px;
font-size: 50px;
padding : 30px;
color:white;
}
#intro{
background-image: url(network.png);
background-size: cover;
width: 550px;
height: 550px;
margin-left: 10px;
margin-right: 250px;
margin-top: 100px;
}
#intro-container{
display: flex;
}
Activate Windows
Go to Settings to activate Windows.
Run
webserver
1networkProject > styleAr.css
190:2 CRLF UTF-8 4 spaces Python 3.12 (1networkProject)
97°F
Mostly cloudy
Search
ENG
2:55 PM
5/10/2024

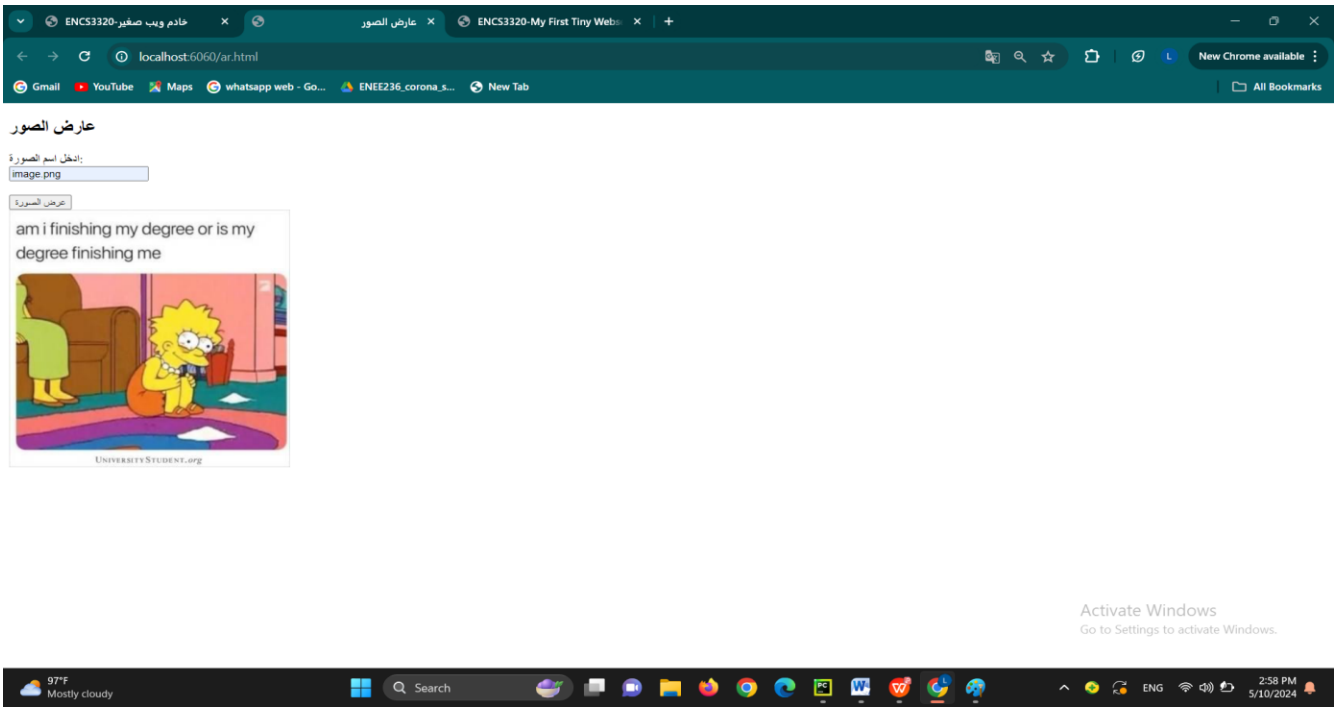
- **Web server**


ENCS3320-خادم ويب صغير
ENCS3320-My First Tiny Webs
localhost:6060/ar
New Chrome available
Gmail
YouTube
Maps
whatsapp web - Go...
ENEE236_corona_s...
New Tab
All Bookmarks
مرحبًا بكم في مساقنا شبكات الحاسوب، ENCS3320-
خادمي الويب الصغير
Activate Windows
Go to Settings to activate Windows.
97°F
Mostly cloudy
Search
ENG
2:56 PM
5/10/2024


ENCS3320-خادم ويب صغير
ENCS3320-My First Tiny Webs
localhost:6060/ar
New Chrome available
Gmail
YouTube
Maps
whatsapp web - Go...
ENEE236_corona_s...
New Tab
All Bookmarks
الاسم: ليليان كراجة
الرقم الجامعي: 1201191
الهوايات:
الرسم
الطبخ
تصوير الصور
المشاريع:
نظام لتوقع حدد المتبرعين بالدم (مادة الذكاء الاصطناعي)
نظام لخوارزميات المجدد (مادة نظم التشغيل)
مشروع لمحاكاة دوائر النبود (مادة الكترونيات)
الاسم: دارين مهنا
الرقم الجامعي: 1202963
الهوايات:
القراءة
البرمجة
التصميم
المشاريع:
عيادة أسنان (مادة قواعد البيانات)
نظام مدرسي للطلاب (مادة هياكل البيانات و الخوارزميات)
آلة حاسبة بسيطة (مادة هياكل البيانات و الخوارزميات)
Activate Windows
Go to Settings to activate Windows.
97°F
Mostly cloudy
Search
ENG
2:56 PM
5/10/2024


ENCS3320-خادم ويب صغير
ENCS3320-My First Tiny Webs
localhost:6060/ar
New Chrome available
Gmail
YouTube
Maps
whatsapp web - Go...
ENEE236_corona_s...
New Tab
All Bookmarks
الاسم: دارين مهنا
الرقم الجامعي: 1202963
الهوايات:
القراءة
البرمجة
التصميم
المشاريع:
عيادة أسنان (مادة قواعد البيانات)
نظام مدرسي للطلاب (مادة هياكل البيانات و الخوارزميات)
آلة حاسبة بسيطة (مادة هياكل البيانات و الخوارزميات)
الاسم: ليليان كراجة
الرقم الجامعي: 1201191
الهوايات:
الرسم
الطبخ
تصوير الصور
المشاريع:
نظام لتوقع عدد المتبرعين بالدم (مادة الذكاء الاصطناعي)
نظام لخوارزميات التجدد (مادة نظم التشغيل)
مشروع لمحاكاة دوائر الديود (مادة الكترونيات)
نموذجي
وثائق المدرسة العالمية 3
Activate Windows
Go to Settings to activate Windows.
97°F
Mostly cloudy
Search
ENG
2:56 PM
5/10/2024


ENCS3320-خادم ويب صغير
عارض الصور
ENCS3320-My First Tiny Webs
localhost:6060/ar.html
New Chrome available
Gmail
YouTube
Maps
whatsapp web - Go...
ENEE236_corona_s...
New Tab
All Bookmarks
عارض الصور
ادخل اسم الصورة:
عرض الصورة
Activate Windows
Go to Settings to activate Windows.
97°F
Mostly cloudy
Search
ENG
2:57 PM
5/10/2024


ENCS3320-خادم ويب صغير
عارض الصور
ENCS3320-My First Tiny Web
localhost:6060/ar.html
New Chrome available
Gmail
YouTube
Maps
whatsapp web - Go...
ENEE236_corona_s...
New Tab
All Bookmarks
عارض الصور
ادخل اسم الصورة:
image.png
عرض الصورة
am i finishing my degree or is my degree finishing me
UNIVERSITYSTUDENT.org
Activate Windows
Go to Settings to activate Windows.
97°F
Mostly cloudy
Search
ENG
2:58 PM
5/10/2024