" بسم الله الرحمن الرحيم "

Faculty of Engineering and Technology.

Electrical and Computer Engineering Department.

ENCS3320-Computer Networks.

Second Semester – 2022|2023.

Project #1: Socket Programing.

****

Students names (Prepared by):

Razan Abdelrahman - 1200531.

Maisam Alaa - 1200650.

Instructor name: Dr. Abdalkarim Awad.

Section: 1.

Date: 14|May.

Contents.

Table of figures.

Part I:

- What are ping, tracert, nslookup, and telnet?

Ping: Ping is a network protocol used to test the availability and responsiveness of a device or server on a network.

Tracert: Short for "traceroute", a command-line networking tool that is used to identify the path that network packets take as they travel from a source to a destination over an IP network.

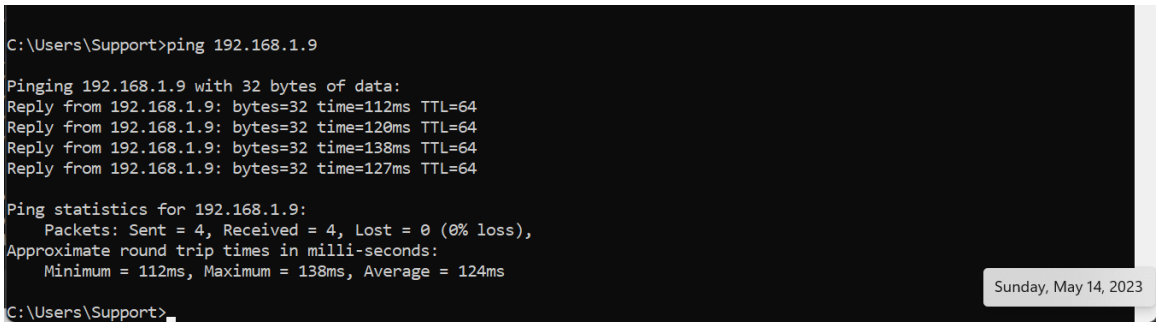
Nslookup: Short for "name server lookup" is a command-line networking tool used to query the Domain Name System (DNS) to obtain information about domain names, IP addresses, and other related information.

Telnet: is a network protocol used for establishing a remote connection between computers over a network.

- Ping a device in the same network.

From laptop to smartphone.

We can see from the figure below that we received a response from (192.168.1.9) when we sent 4 packets where all packets have the same TTL (time to live), all packets are received with different delays and the average is 124 ms.



```
C:\Users\Support>ping 192.168.1.9

Pinging 192.168.1.9 with 32 bytes of data:
Reply from 192.168.1.9: bytes=32 time=112ms TTL=64
Reply from 192.168.1.9: bytes=32 time=120ms TTL=64
Reply from 192.168.1.9: bytes=32 time=138ms TTL=64
Reply from 192.168.1.9: bytes=32 time=127ms TTL=64

Ping statistics for 192.168.1.9:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 112ms, Maximum = 138ms, Average = 124ms

C:\Users\Support>
```

Sunday, May 14, 2023

Figure 1 Pink a device.

- Ping www.harvard.edu.

We can see from the figure below that we received a response from [199.232.82.133] when we sent 4 packets where all packets have the same TTL (time to live), all packets are received with different delays and the average is 67 ms. Also we can notice that the time to obtain a response from harvard.edu takes less than a response from a smartphone in the same network of my laptop.

```
C:\Users\Support>ping www.harvard.edu

Pinging pantheon-systems.map.fastly.net [199.232.82.133] with 32 bytes of data:
Reply from 199.232.82.133: bytes=32 time=68ms TTL=57
Reply from 199.232.82.133: bytes=32 time=68ms TTL=57
Reply from 199.232.82.133: bytes=32 time=67ms TTL=57
Reply from 199.232.82.133: bytes=32 time=67ms TTL=57

Ping statistics for 199.232.82.133:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 67ms, Maximum = 68ms, Average = 67ms

C:\Users\Support>
```

Sunday, May 14, 2023

Figure 2 Ping a website.

- Tracert www.harvard.edu.

This command sends 3 messages for every router and waits the response from the router, it continues in this process until it reaches the chosen IP. We can see from the figure that there is 3 measurements from each router. When we go down in the lines, we will see the 3 measurements increases because the next router go further.

```
C:\Users\Support>tracert www.harvard.edu

Tracing route to pantheon-systems.map.fastly.net [146.75.122.133]
over a maximum of 30 hops:

  1     1 ms     1 ms    <1 ms  192.168.1.1
  2    38 ms   135 ms    15 ms  172.16.250.145
  3    34 ms    18 ms    27 ms  172.16.250.13
  4    73 ms    78 ms    74 ms  63-218-13-193.static.pccwglobal.net [63.218.13.193]
  5    72 ms    72 ms    80 ms  Hun0-0-0-1.br01.ldn01.pccwbtn.net [63.218.12.46]
  6   240 ms   375 ms   396 ms  ldn-b2-link.ip.twelve99.net [80.239.193.134]
  7   164 ms   319 ms   135 ms  ldn-bb1-link.ip.twelve99.net [62.115.122.188]
  8   139 ms    95 ms    98 ms  prs-bb1-link.ip.twelve99.net [62.115.135.25]
  9   108 ms    86 ms    81 ms  ffm-bb1-link.ip.twelve99.net [62.115.123.12]
 10    88 ms    96 ms    84 ms  ffm-b5-link.ip.twelve99.net [62.115.114.89]
 11    85 ms    92 ms    85 ms  fastly-ic-373938.ip.twelve99-cust.net [62.115.187.105]
 12    85 ms    85 ms   132 ms  146.75.122.133

Trace complete.

C:\Users\Support>
```

Sunday, May 14, 2023

Figure 3 Tracert a website.

- Nslookup www.harvard.edu

We can see that it prints the IP address corresponding to the host which is my laptop's IP address, and prints the name and addresses of the server which is the host that we sent a probe.

```
C:\Users\Support>nslookup www.harvard.edu
Server:  r3.mada.ps
Address:  185.17.235.133

Non-authoritative answer:
Name:    pantheon-systems.map.fastly.net
Addresses:  2a04:4e42:54::645
          146.75.122.133
Aliases:  www.harvard.edu


C:\Users\Support>
```

Sunday, May 14, 2023

Figure 4 Nslook up a website.

****

Part II:

❖ UDP server.

■ Code.

```python
#import the neede librarires.
import socket
import time

# Create a UDP socket.
server_socket = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)

# Bind the socket to a specific address and port.
server_address = ('', 8855)
server_socket.bind(server_address)

# Dictionary to store the last received message from each client.
client_messages = {}

#This message to know the user that server start working.
print('\nServer is listening on port 8855...\n')

#Server name.
print('Server Razan Odeh.')

index = 0
while True:
    # Receive a message and the client address.
    data, client_address = server_socket.recvfrom(1024)
    message = data.decode()

    # Get the client name from the message.
    client_name = message.split(",")[0]

    # Update the last received message from the client.
    client_messages[client_address] = message


    index += 1

    # Print the last received message from each client.
    for address, msg in client_messages.items():
        name = msg.split(",")[0]
        last_received = time.strftime('%H:%M:%S')
        print(f"{index} - {name}: Last received message at {last_received}")
    print("====================================")

# Close the socket.
server_socket.close()
```

Figure 5 Server code.

Result.

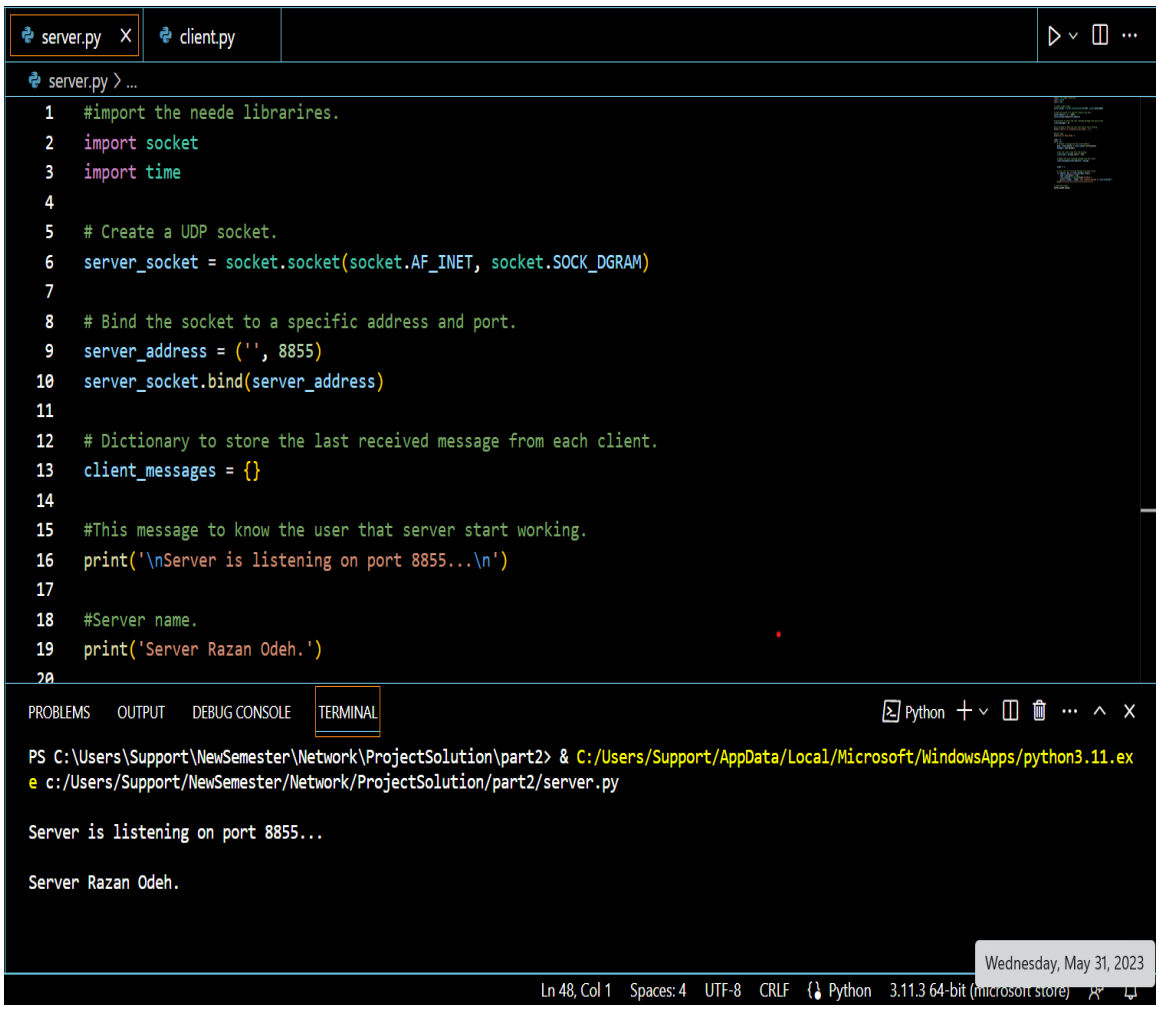


Figure 6 Server output.

****

❖ UDP client.

■ Code.

```python
#Import the needed libraries.
import socket
import time

# Create a UDP socket.
client_socket = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)

# Enable broadcasting.
client_socket.setsockopt(socket.SOL_SOCKET, socket.SO_BROADCAST, 1)

# Set the broadcast address and port.
broadcast_address = ("172.19.255.255", 8855)

while True:
    client_name = "Maisam Alaa"
    # Construct the message to be sent.
    message = f"{client_name},{socket.gethostbyname(socket.gethostname())}"

    # Send the message to the broadcast address.
    client_socket.sendto(message.encode(), broadcast_address)

    # Print the sent message and current time when the message has sent.
    print(f"Sent message: {message} at {time.strftime('%H:%M:%S')}")

    # Wait for 2 seconds before sending the next message.
    time.sleep(2)
```

Figure 7 Client Code.

■ Result.

```python
#import the neede librarires.
import socket
import time

# Create a UDP socket.
server_socket = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)

# Bind the socket to a specific address and port.
server_address = ('', 8855)
server_socket.bind(server_address)

# Dictionary to store the last received message from each client.
client_messages = {}

#This message to know the user that server start working.
print('\nServer is listening on port 8855...\n')

#Server name.
print('Server Razan Odeh.')
```

```
PS C:\Users\Support\NewSemester\Network\ProjectSolution\part2> python client.py
Sent message: Maisam Alaa,172.19.34.136 at 13:09:05
Sent message: Maisam Alaa,172.19.34.136 at 13:09:07
Sent message: Maisam Alaa,172.19.34.136 at 13:09:09
Sent message: Maisam Alaa,172.19.34.136 at 13:09:11
Sent message: Maisam Alaa,172.19.34.136 at 13:09:13
Sent message: Maisam Alaa,172.19.34.136 at 13:09:15
Sent message: Maisam Alaa,172.19.34.136 at 13:09:17
Sent message: Maisam Alaa,172.19.34.136 at 13:09:19
```

Figure 8 Client Output

Final result.

After running the server code and the client code to send messages from server to client,

The output was like that:

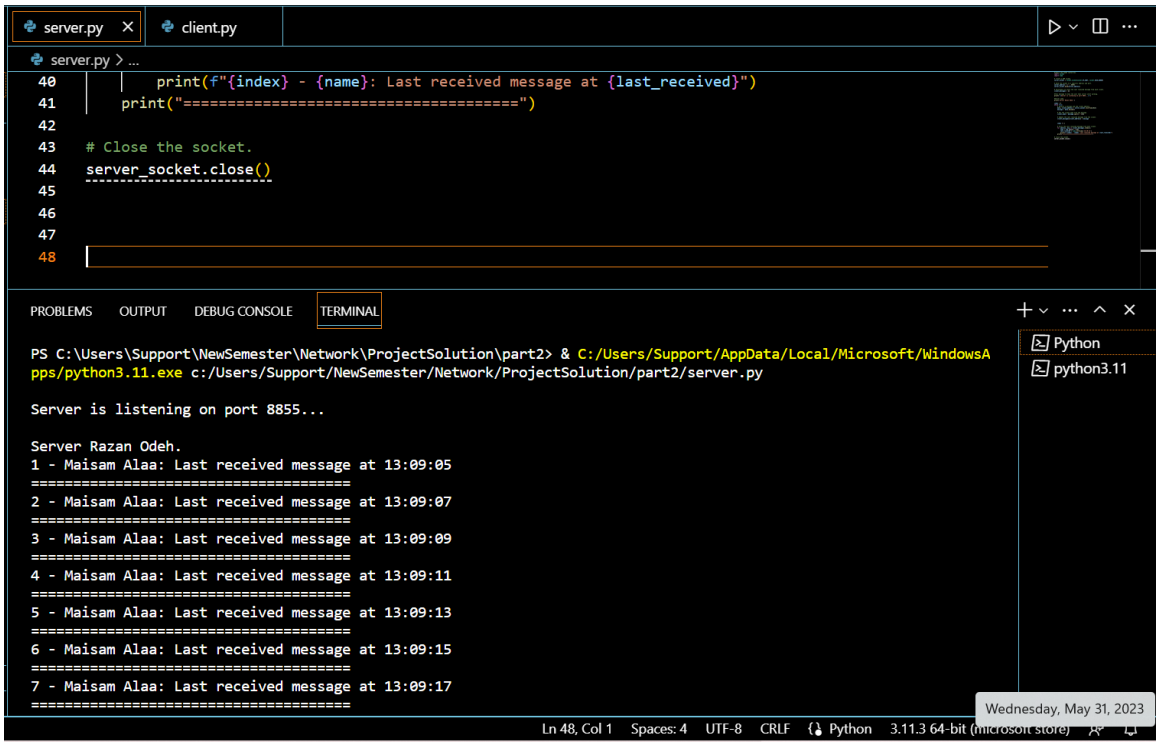


Figure 9 Final output.

The messages have received from the client 'Maisam Alaa' to the server 'Razan Odeh',

The time difference between each message and the other is 2sec.

****

Part III:

Using socket programming, implement a simple but a complete web server in go, python, java or C that is listening on port 9977.

- Running the server:

Figure 10 main.py

Then, the server is starting and ready to receive requests.

****

- Web main page in English:

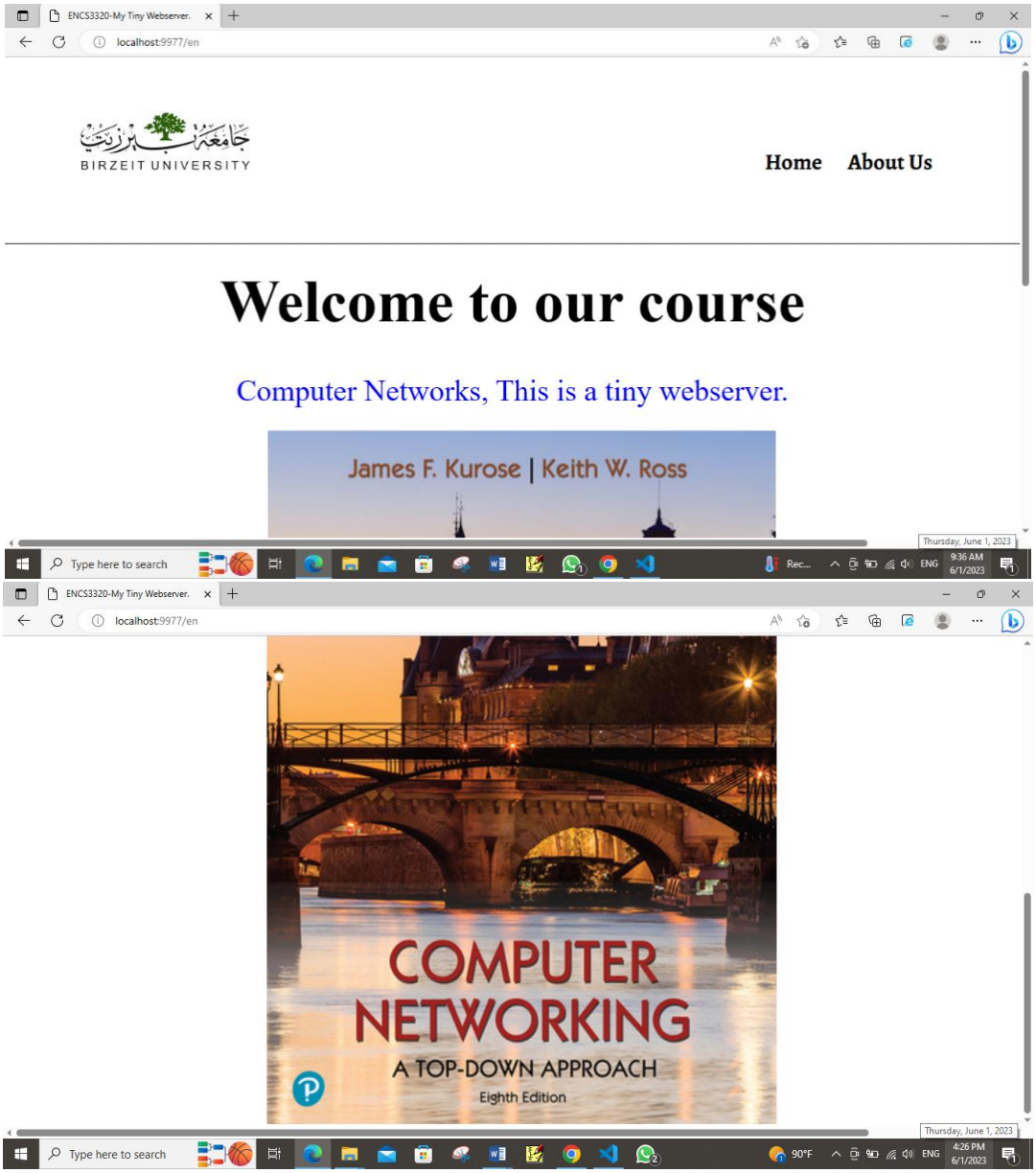


Figure 11 main page in English.

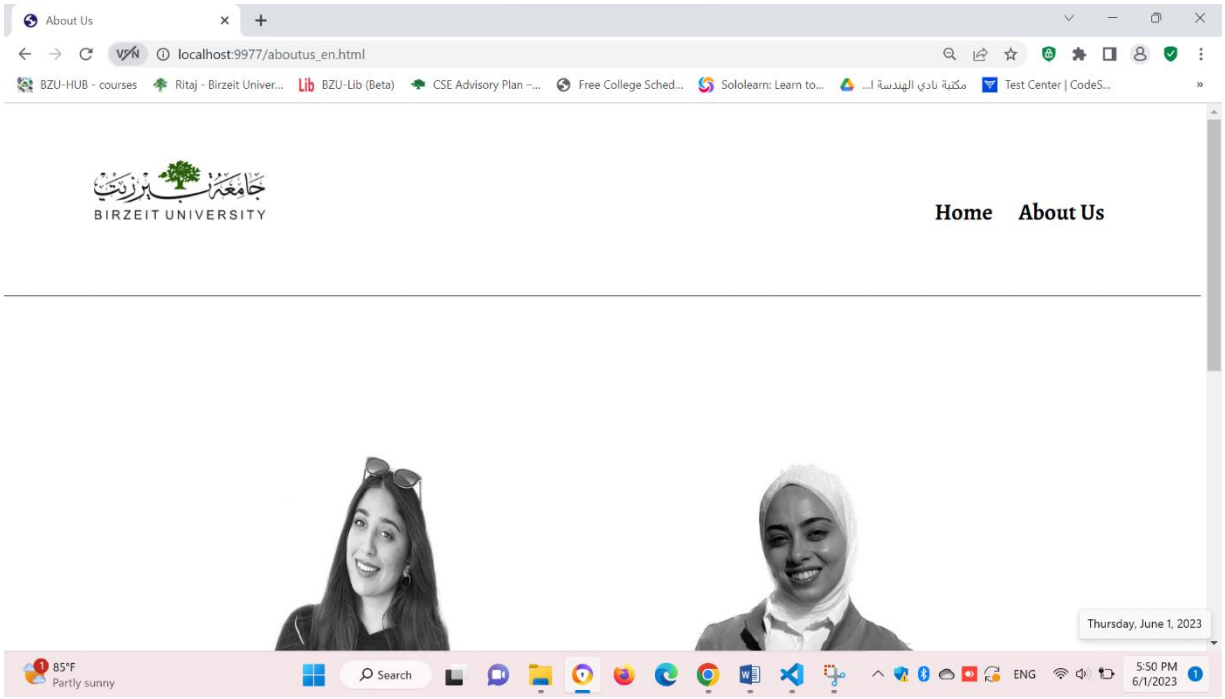
****

- Web main page in Arabic:

Figure 12 main page in Arabic.

****

- Introduction page for team members in English:



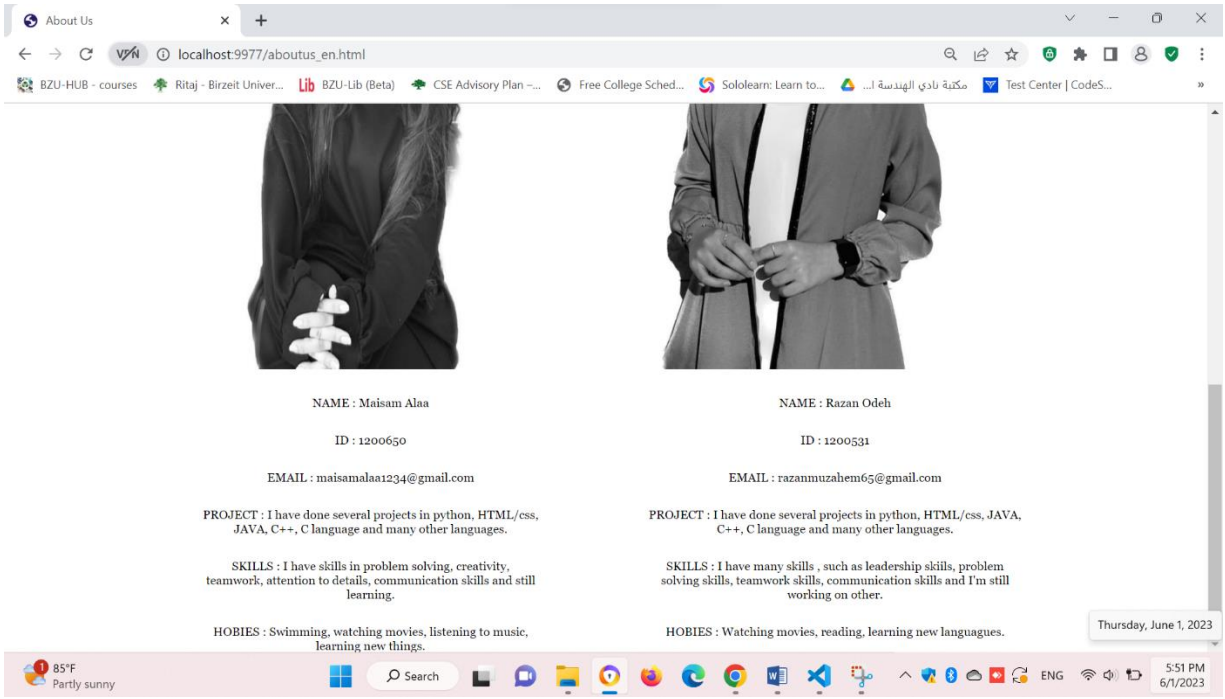


Figure 13 about us page in English.

****

- Introduction page for team members in Arabic:

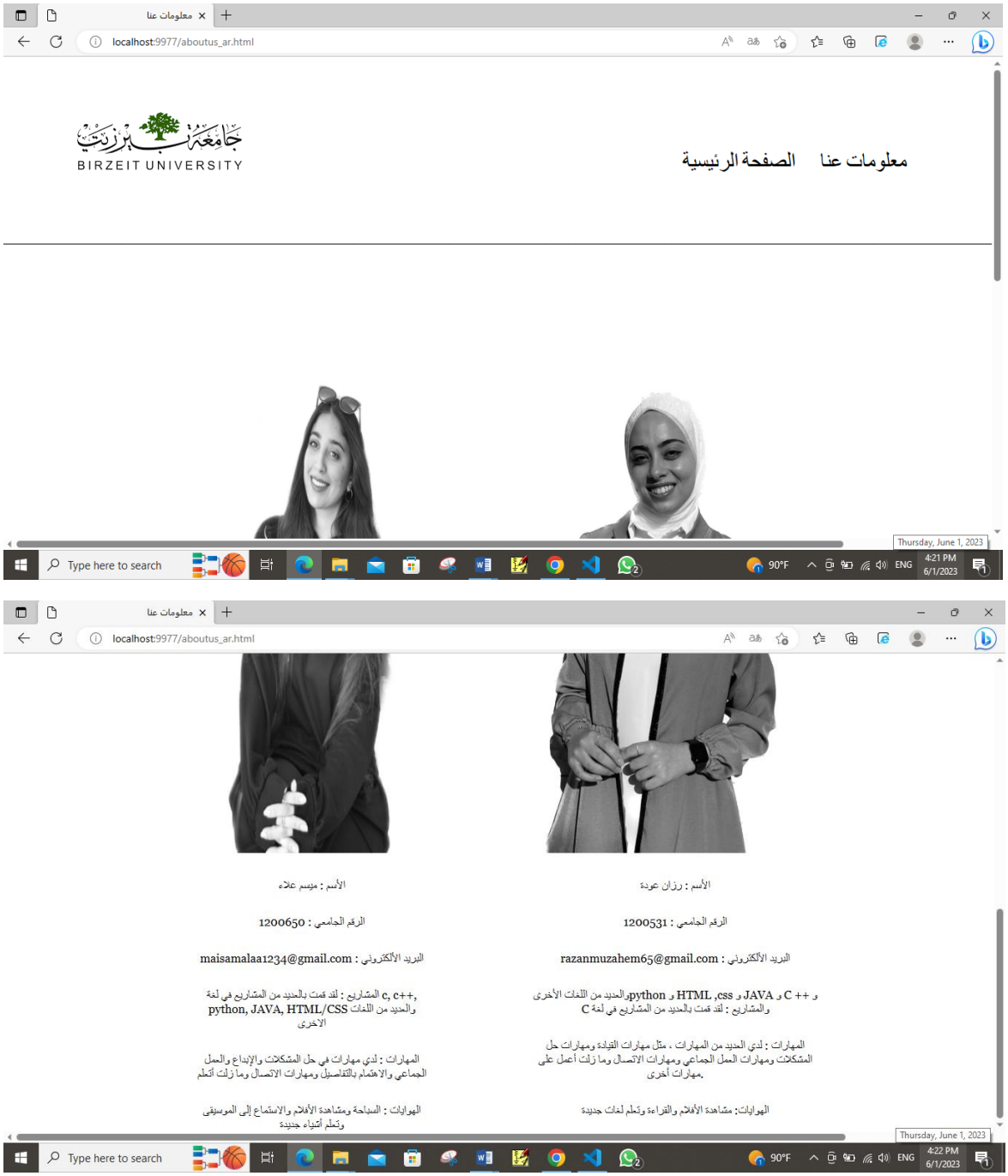
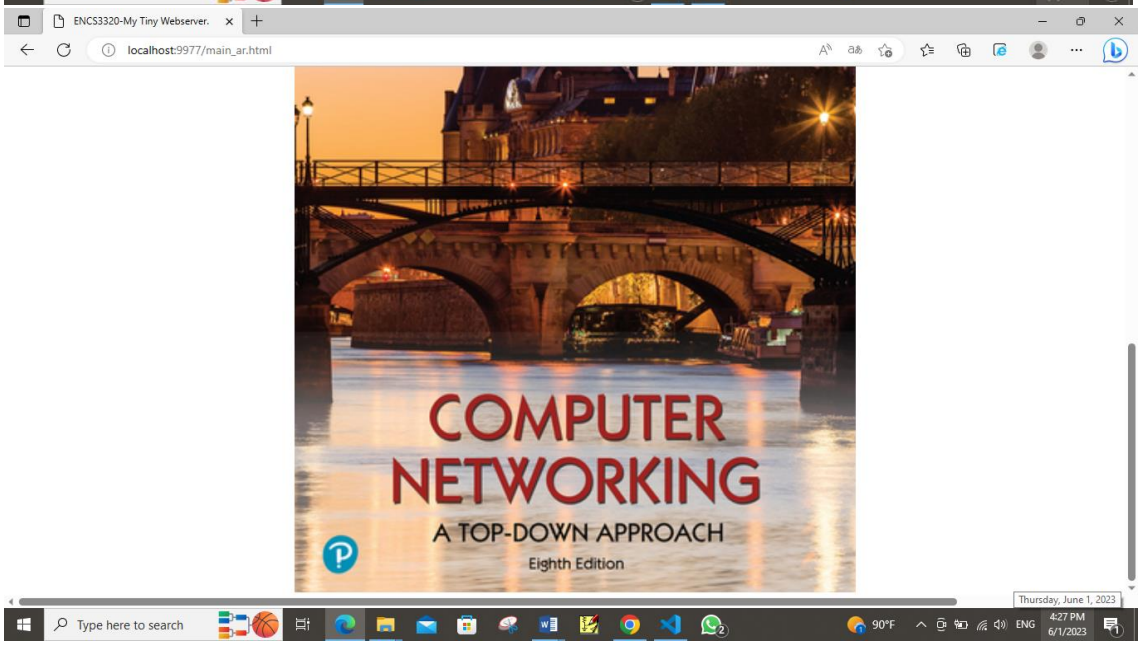


Figure 14 about us page in Arabic.

****

Then, we sent some requests to the server, such as:

- ✓ <u>HTML Page:</u>

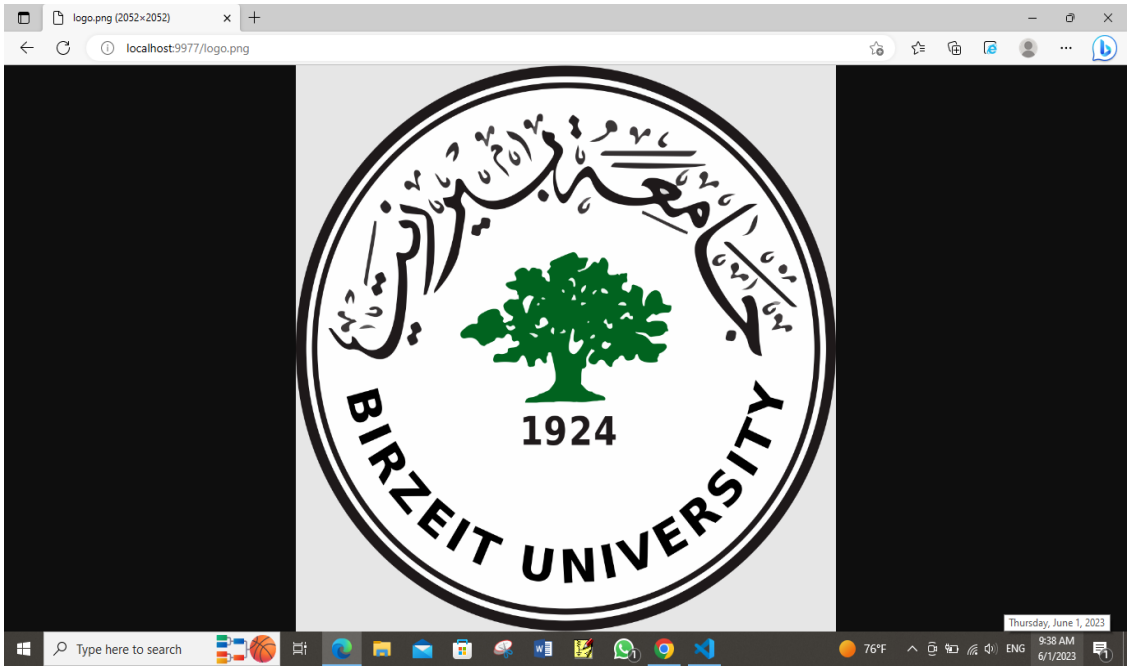


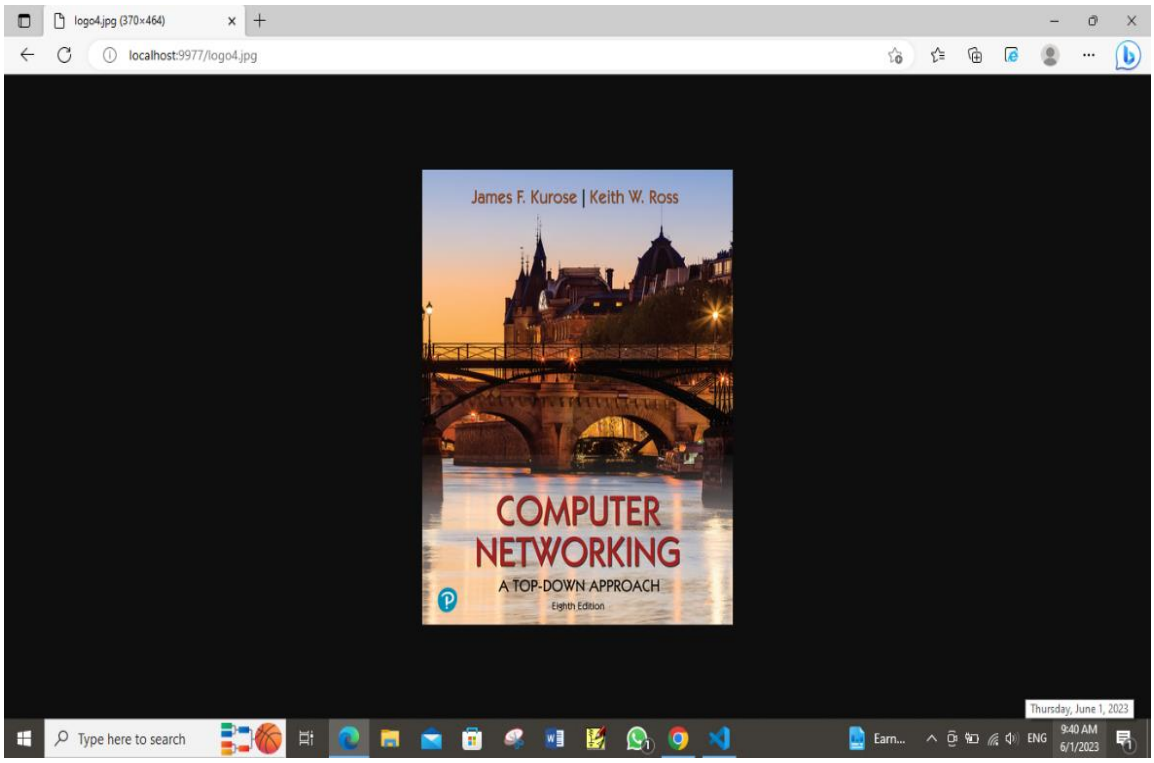
✓ <u>CSS Page:</u>

```
@import url('https://fonts.googleapis.com/css2?family=Alegreya:wght@700&display=swap');
/* font-family: 'Alegreya', serif;   font-family: 'Dancing Script', cursive;
*/
@import url('https://fonts.googleapis.com/css2?family=Alegreya:wght@700&family=Dancing+Script:wght@600&display=swap');
*{
    margin :0;
    padding : 0;
}
body{
    background-color: white;

}
.container{
    width: 85%;
    margin: auto;
}
.navbar{
    overflow: hidden;
}
.navbar .logo{
    float: left;

}
.navbar .navitem{
    float: right;

}
.navbar .logo .img{
    width: 85%;
    margin: auto;
    padding-top: 0px;

}
.navbar .navitem li{
    display: inline-block;
    padding: 15px 0;
    padding-top: 120px;

}
.navbar .navitem li a{
    color: black;
    text-decoration: none;
    padding: 0 15px;
```

```
}
.navbar .navitem li{
    display: inline-block;
    padding: 15px 0;
    padding-top: 120px;

}
.navbar .navitem li a{
    color: black;
    text-decoration: none;
    padding: 0 15px;
    text-transform: capitalize;
    font-family: 'Alegreya', serif;
    font-size: 30px;


}
p{
    color: blue;
    text-align: center;
    padding: 0;
    margin: 0;
    margin-bottom: 0;
    font-size: 40px;
}
h1{
    text-align: center;
    padding: 0;
    margin: 0;
    margin-bottom: 0;
    padding-bottom: 50px;
    font-size: 80px;
    padding-top: 30px;
}
.newlogo img{
    width: 50%;
    margin: auto;
    padding-top: 30px;
    padding-left: 350px;
}
```

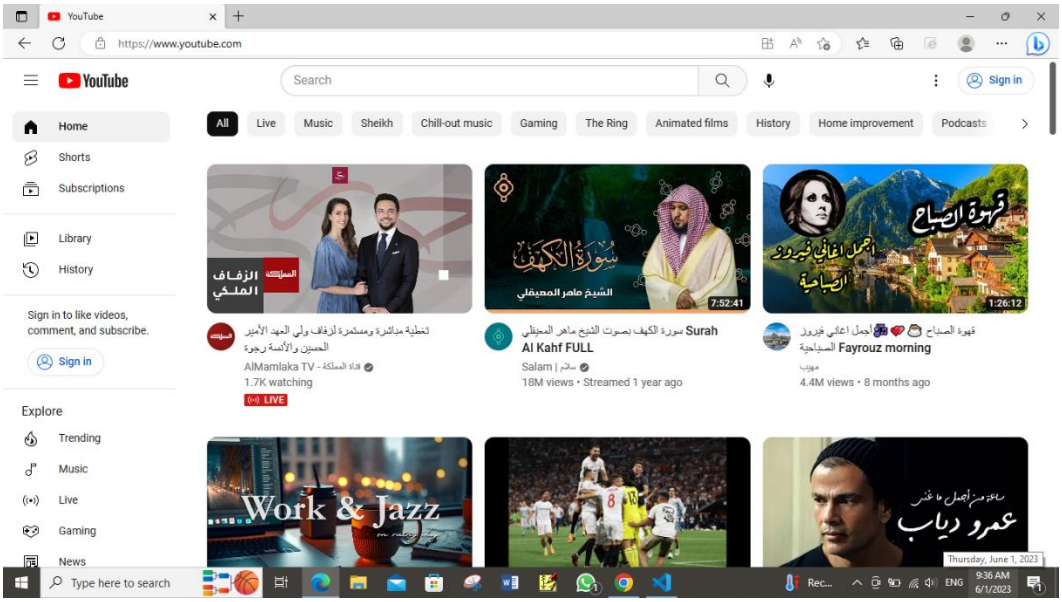
✓ <u>Request images with the extension png, jpg:</u>



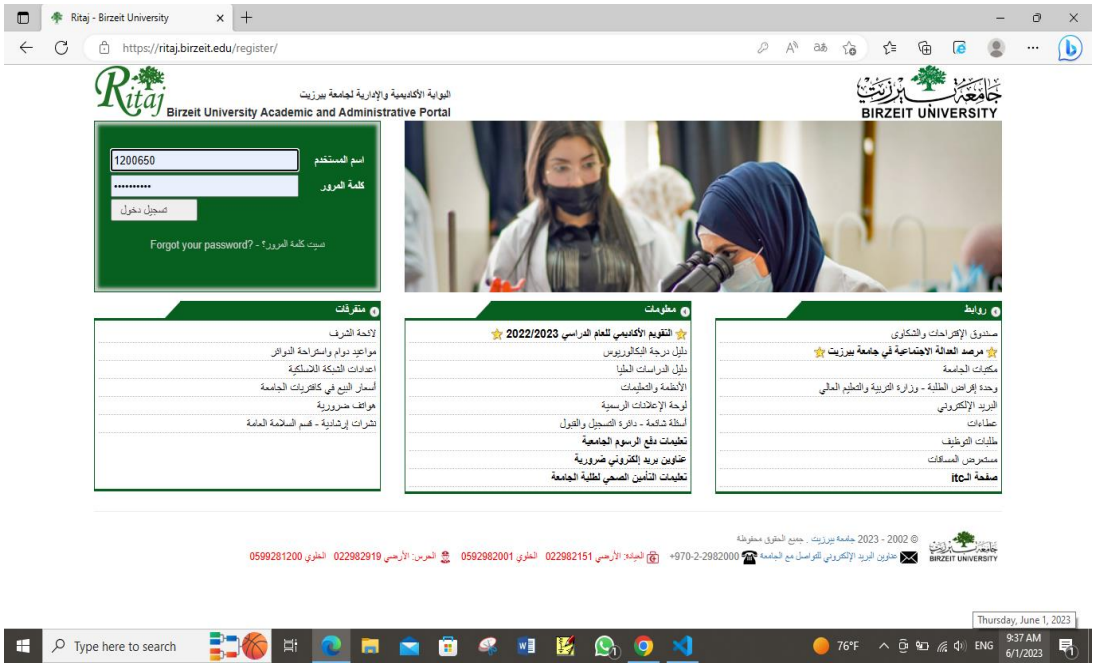


✓ <u>Request different websites:</u>

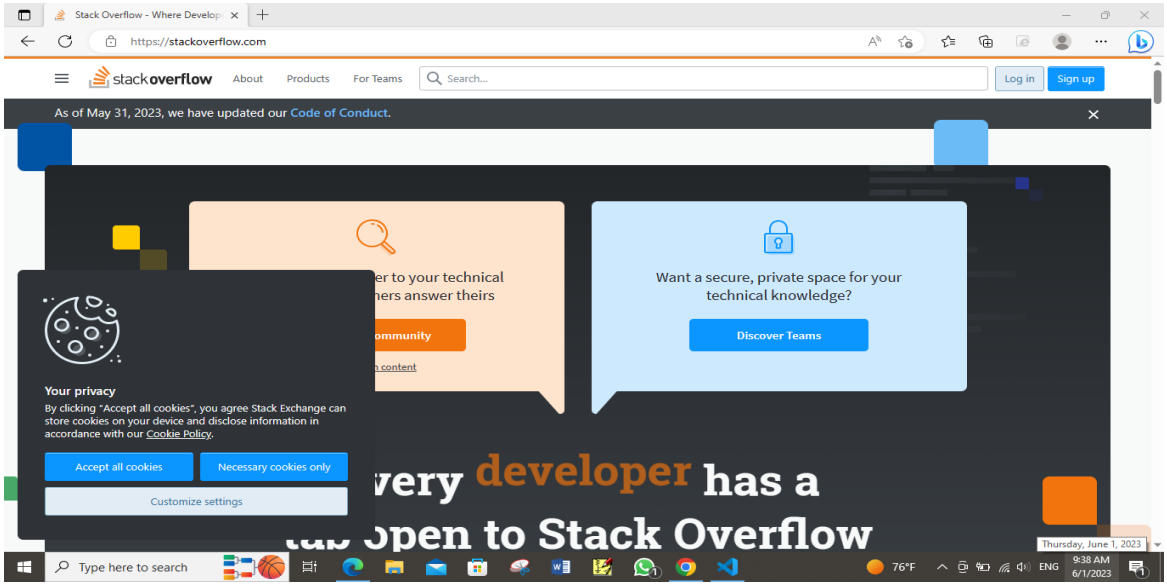
Yt| https://www.youtube.com/



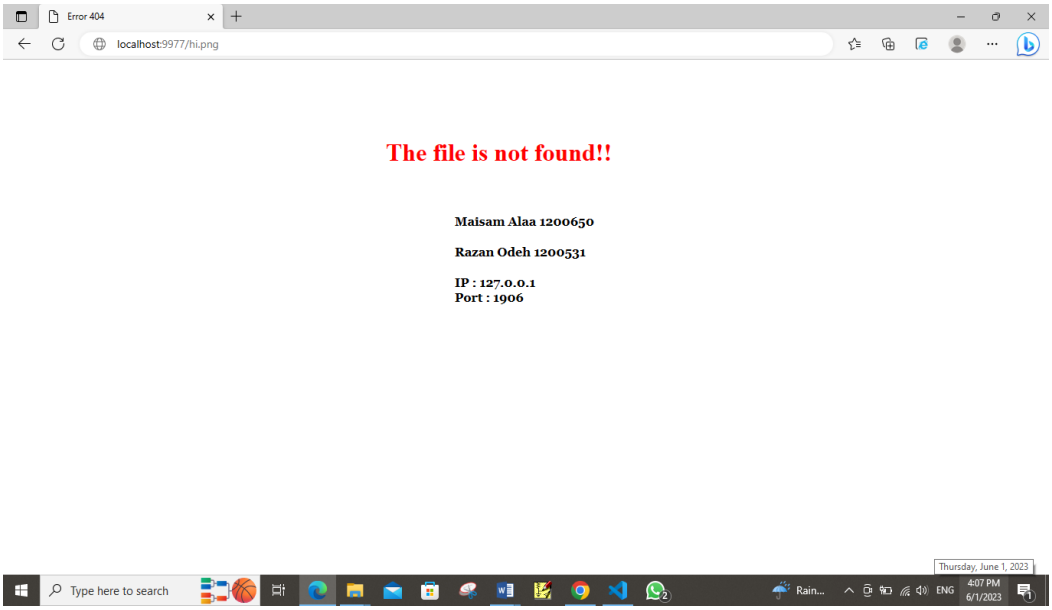
Rt| https://ritaj.birzeit.edu/register/:

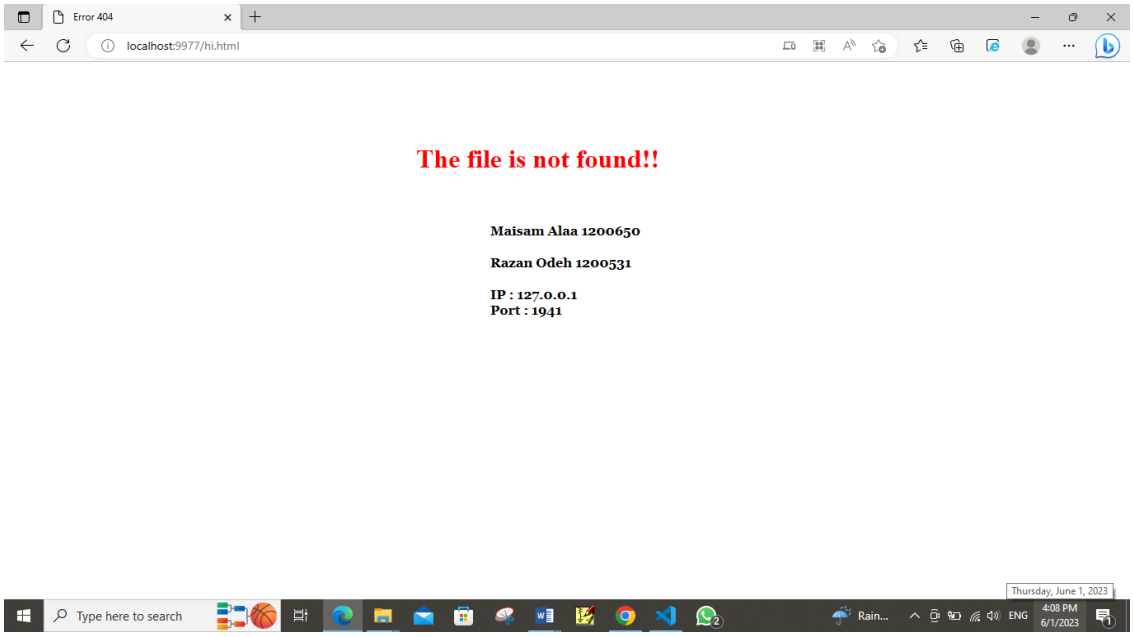


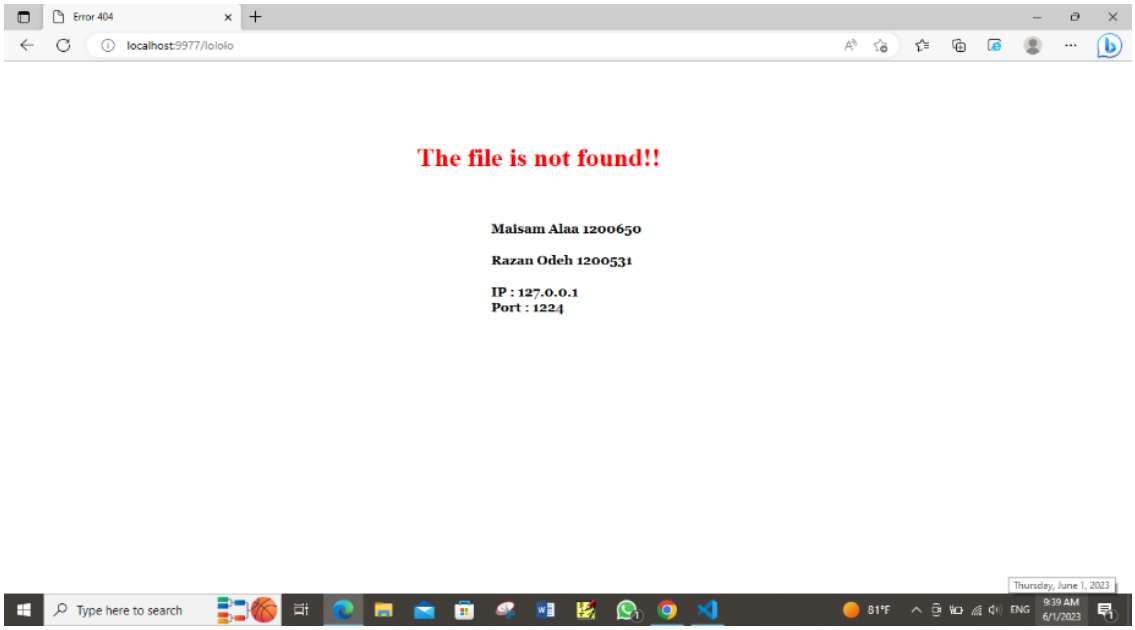
So| https://stackoverflow.com/



✓ Request some pages and photos that not found:

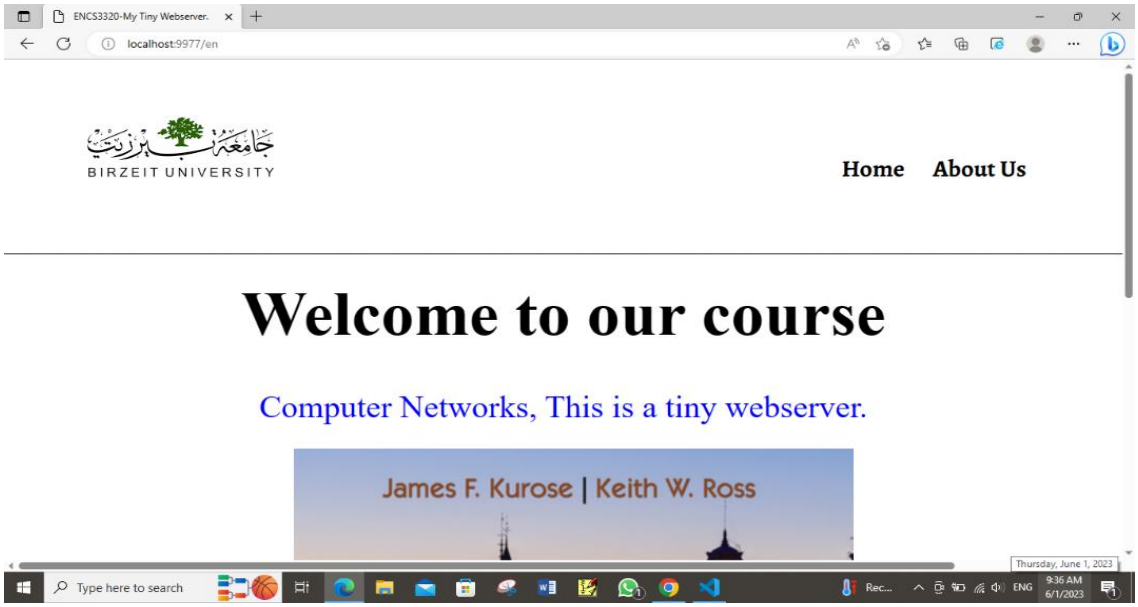



Error 404
localhost:9977/hi.html
The file is not found!!
Maisam Alaa 1200650
Razan Odeh 1200531
IP : 127.0.0.1
Port : 1941
Type here to search
Thursday, June 1, 2023
4:08 PM
6/1/2023


Error 404
localhost:9977/lololo
The file is not found!!
Maisam Alaa 1200650
Razan Odeh 1200531
IP : 127.0.0.1
Port : 1224
Type here to search
Thursday, June 1, 2023
9:39 AM
6/1/2023

****

- The webserver from different computer:



****