Programming Assignment 2 – CSE232

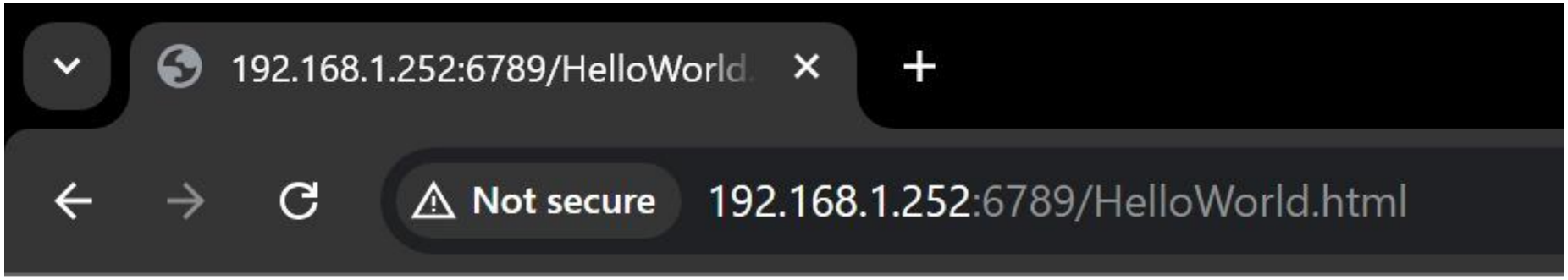
Palak Bhardwaj 2022344
Yashovardhan Singhal 2022591

INDRAPRASTHA INSTITUTE *of*
INFORMATION TECHNOLOGY
DELHI

Part (a) – Case (File present at Server)

Initially both client and server are run on the same machine, the server in this case is the localhost.

Then the client and server are run on two different devices connected to the same network.
The client sends a request, from a browser, for the HelloWorld.html via a valid URL, once the server accepts the request, the client can get the file at that server.



Output on the server side

This message confirms that the web server has successfully started and is now listening for incoming connections on port 6789.

The server displays this message when the client is able to successfully is able to get the HTML file present at the server.

The content of the request HTML file is printed which confirms the presence of the file.

```
PS C:\Users\Yashovardhan Singhal\Documents\Sem5\CN\Assignment2\part1> python TCP.py
Web server is up on port:  6789
Requested file: HelloWorld.html
<a>Palak and Yashovardhan 's CN assignment 2</a>
Ready to serve...
```

Part (a) – Case (File not present at Server)

If the client tries to access a file which is not present at the server, the following output is displayed by the server.

This message confirms that the web server has successfully started and is now listening for incoming connections on port 6789.
"Ready to serve..." (again):

This message reappears after the file-not-found event, showing that the server continues running and is still available to handle future requests.

```
PS C:\Users\Yashovardhan Singhal\Documents\Sem5\CN\Assignment2\part1> python TCP.py
Web server is up on port:  6789
Ready to serve...
Requested file: GoodbyeWorld.html
File Not Found
Ready to serve...
```

Need for Multithreading

The current server cannot handle multiple requests concurrently, multiple requests sent together are handled sequentially.

```
PS C:\Users\Yashovardhan Singhal\Documents\Sem5\CN\Assignment2\part1> python TCPdelay.py
Web server is up on port: 6789
Ready to serve...
Simulating a delay for request from ('127.0.0.1', 50491)
Received message: GET /HelloWorld.html HTTP/1.1
User-Agent: Mozilla/5.0 (Windows NT; Windows NT 10.0; en-IN) WindowsPowerShell/5.1.22621.2506
Host: localhost:6789


Requested file: HelloWorld.html
<a>Palak and Yashovardhan 's CN assignment 2</a>
Ready to serve...
Simulating a delay for request from ('127.0.0.1', 50492)
Received message: GET /HelloWorld.html HTTP/1.1
User-Agent: Mozilla/5.0 (Windows NT; Windows NT 10.0; en-IN) WindowsPowerShell/5.1.22621.2506
Host: localhost:6789


Requested file: HelloWorld.html
<a>Palak and Yashovardhan 's CN assignment 2</a>
Ready to serve...
```

Part(B) Multithreading at Server end

This shows the presence of multiple requests being handled together instead of sequentially being handled as in the previous case.

```
PS C:\Users\Yashovardhan Singhal\Documents\Sem5\CN\Assignment2\part2> python Multi.py
Web server is up and running on port: 6789
Simulating a delay for request from ('127.0.0.1', 50627)
Simulating a delay for request from ('127.0.0.1', 50629)
Connection established with ('127.0.0.1', 50627)
Received message from ('127.0.0.1', 50627):
GET /HelloWorld.html HTTP/1.1
User-Agent: Mozilla/5.0 (Windows NT; Windows NT 10.0; en-IN) WindowsPowerShell/5.1.22621.2506
Host: localhost:6789
Connection: Keep-Alive


Connection established with ('127.0.0.1', 50629)
Received message from ('127.0.0.1', 50629):
GET /HelloWorld.html HTTP/1.1
User-Agent: Mozilla/5.0 (Windows NT; Windows NT 10.0; en-IN) WindowsPowerShell/5.1.22621.2506
Host: localhost:6789
Connection: Keep-Alive
```

Client Code 1 (2 clients)

2 clients running concurrently via multithreading.

```
PS C:\Users\Yashovardhan Singhal\Documents\Sem5\CN\Assignment2>
 curl http://localhost:6789/HelloWorld.html


StatusCode        : 200
StatusDescription : OK
Content           : <a>'HIIIIII'</a>
RawContent        : HTTP/1.1 200 OK
                    Connection: close
                    Content-Type: text/html

                    <a>'HIIIIII'</a>
Forms             : {}
Headers           : {[Connection, close], [Content-Type, text/html]}
Images            : {}
InputFields       : {}
Links             : {}
ParsedHtml        : mshtml.HTMLDocumentClass
RawContentLength  : 16
```

Client Code 2 (2 clients)

Client 2 accessing the server simultaneously with client1.

```
PS C:\Users\Yashovardhan Singhal\Documents\Sem5\CN\Assignment2> curl http://localhost:6789/Hello
World.html


StatusCode        : 200
StatusDescription : OK
Content           : <a>'HIIIIII'</a>
RawContent        : HTTP/1.1 200 OK
                    Connection: close
                    Content-Type: text/html

                    <a>'HIIIIII'</a>
Forms             : {}
Headers           : {[Connection, close], [Content-Type, text/html]}
Images            : {}
InputFields       : {}
Links             : {}
ParsedHtml        : mshtml.HTMLDocumentClass
RawContentLength  : 16
```

Part(c) Client End

The client sends HTTP request for HelloWorld.html from server in part a.

```
PS C:\Users\Yashovardhan Singhal\Documents\Sem5\CN\Assignment2> python client.py 127.0.0.1 6789
HelloWorld.html
Sending HTTP request to 127.0.0.1:6789 for file HelloWorld.html...

GET /HelloWorld.html HTTP/1.1
Host: 127.0.0.1
Connection: close


Server response:

HTTP/1.1 200 OK
Content-Type: text/html

<a>Palak and Yashovardhan 's CN assignment 2</a>

PS C:\Users\Yashovardhan Singhal\Documents\Sem5\CN\Assignment2>
```

Part (c) server end

This is server used in part (a), this is what the server used in part(a) displays

```
Ready to serve...
Simulating a delay for request from ('127.0.0.1', 50928)
Received message: GET /HelloWorld.html HTTP/1.1
Host: 127.0.0.1
Connection: close



Requested file: HelloWorld.html
<a>Palak and Yashovardhan 's CN assignment 2</a>
Ready to serve...
```

Part(c) Client End

This is client side when requested from server in part(b).

```
PS C:\Users\Yashovardhan Singhal\Documents\Sem5\CN\Assignment2> python client.py 127.0.0.1 6789
HelloWorld.html
Sending HTTP request to 127.0.0.1:6789 for file HelloWorld.html...

GET /HelloWorld.html HTTP/1.1
Host: 127.0.0.1
Connection: close


Server response:

HTTP/1.1 200 OK
Content-Type: text/html
Connection: close

<a>'HIIIIII'</a>
PS C:\Users\Yashovardhan Singhal\Documents\Sem5\CN\Assignment2>
```

Server End of part(c)

CLient.py code is requesting from server of part b

```
Web server is up and running on port: 6789
Simulating a delay for request from ('127.0.0.1', 50990)
Connection established with ('127.0.0.1', 50990)
Received message from ('127.0.0.1', 50990):
GET /HelloWorld.html HTTP/1.1
Host: 127.0.0.1
Connection: close
```

Server Part (b) –2 client codes are run together

Both the requests are handled concurrently by the server

```
Simulating a delay for request from ('127.0.0.1', 51029)
Simulating a delay for request from ('127.0.0.1', 51030)
Connection established with ('127.0.0.1', 51029)
Received message from ('127.0.0.1', 51029):
GET /HelloWorld.html HTTP/1.1
Host: 127.0.0.1
Connection: close


Connection established with ('127.0.0.1', 51030)
Received message from ('127.0.0.1', 51030):
GET /HelloWorld.html HTTP/1.1
Host: 127.0.0.1
Connection: close
```

Server Part (a) – 2 client codes run together

We can see that even though codes were run concurrently the server executes them sequentially, when the client is run.

```
PS C:\Users\Yashovardhan Singhal\Documents\Sem5\CN\Assignment2\part1> python TCPdelay.py
Web server is up on port: 6789
Ready to serve...
Simulating a delay for request from ('127.0.0.1', 51075)
Received message: GET /HelloWorld.html HTTP/1.1
Host: 127.0.0.1
Connection: close


Requested file: HelloWorld.html
<a>Palak and Yashovardhan 's CN assignment 2</a>
Ready to serve...
Simulating a delay for request from ('127.0.0.1', 51076)
Received message: GET /HelloWorld.html HTTP/1.1
Host: 127.0.0.1
Connection: close


Requested file: HelloWorld.html
<a>Palak and Yashovardhan 's CN assignment 2</a>
Ready to serve...
```