

Name – Dona Maria Jose
UTA ID – 1001232757
Project 1 – Web Proxy Server - Caching
Language – Python
Tool – PyCharm Professional Edition

Python modules used:

socket, sys, thread, logging, time, requests

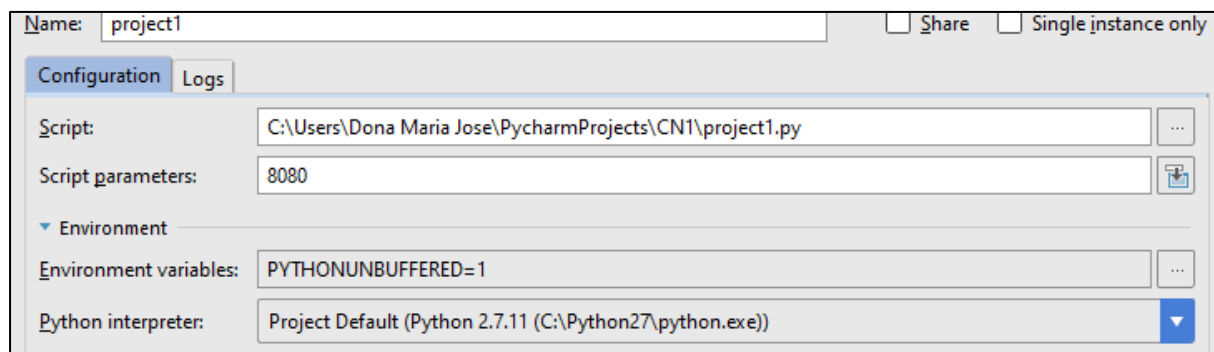
Steps to run the code: -

If you are running with command prompt:

1. Open command prompt
2. Move to the folder where the program is stored, using cd command
cd D:\UTA Notes\Fall 16\CN1\Assignment 1
3. Run the program using python command. Please note to give the port number as argument. If not, port number will be set as 8080, by default.
python project1.py 8080

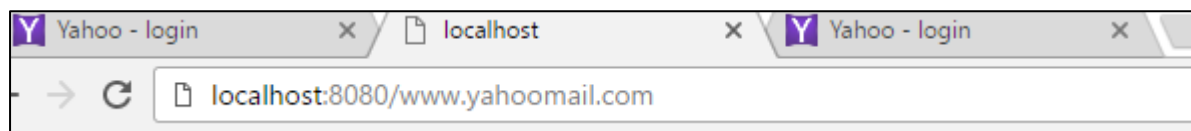
If you are executing the program using PyCharm:

1. Go to Run-> Edit Configurations-> Script Parameters and enter port number as first argument of the code.
2. Click Apply and Ok.
3. Execute the program and see the terminal for response



Continuation:

4. Go to your browser and open the web page using the following format:
localhost: portNo/www.site.com

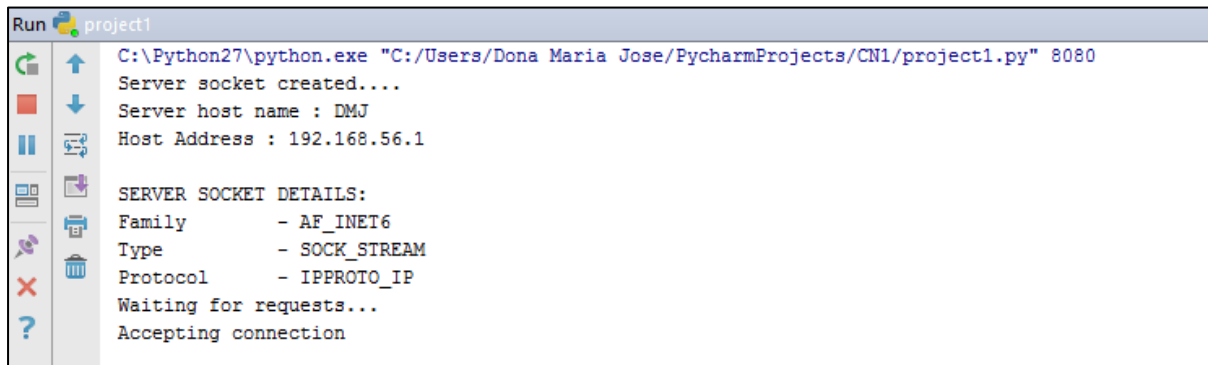


The program was tested with two site:

1. localhost:8080/www.yahoo.com
2. localhost:8080/www.amazon.com

DEAILED DESCRIPTION

- When the program is executed it will create the socket and wait for client request

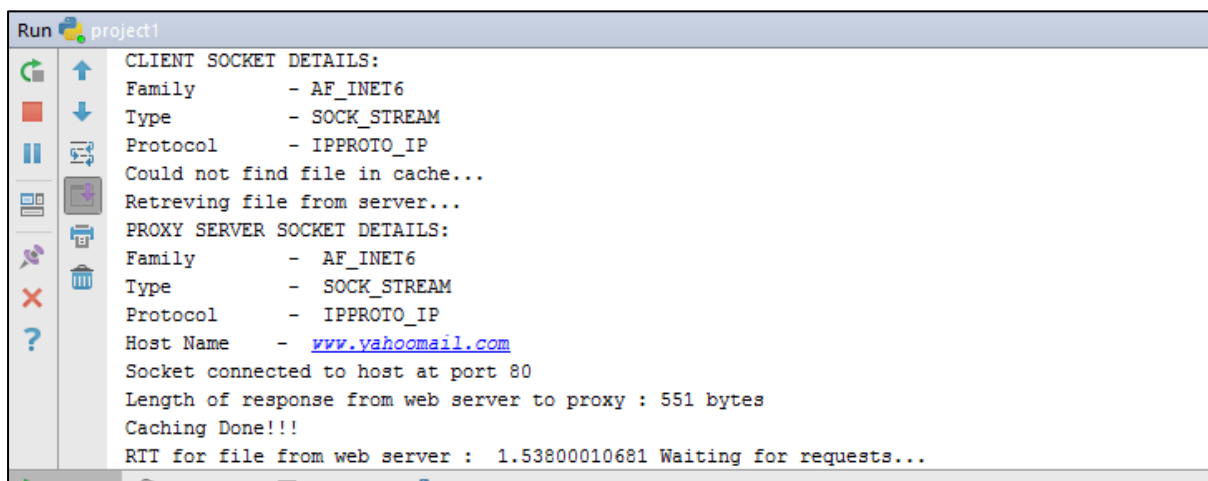


```
Run project1
C:\Python27\python.exe "C:/Users/Dona Maria Jose/PycharmProjects/CN1/project1.py" 8080
Server socket created....
Server host name : DMJ
Host Address : 192.168.56.1

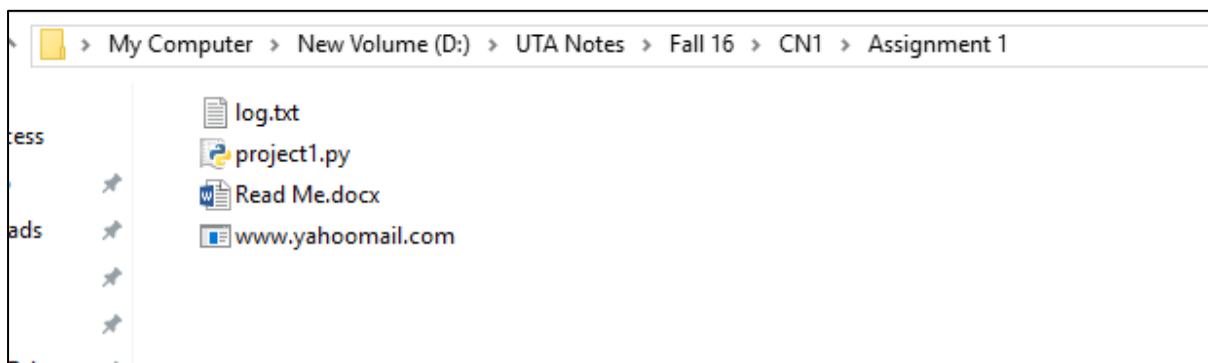
SERVER SOCKET DETAILS:
Family      - AF_INET6
Type        - SOCK_STREAM
Protocol    - IPPROTO_IP
Waiting for requests...
Accepting connection
```

It also displays the socket details like family, type, protocol, etc. All these details are printed for each socket created in the program.

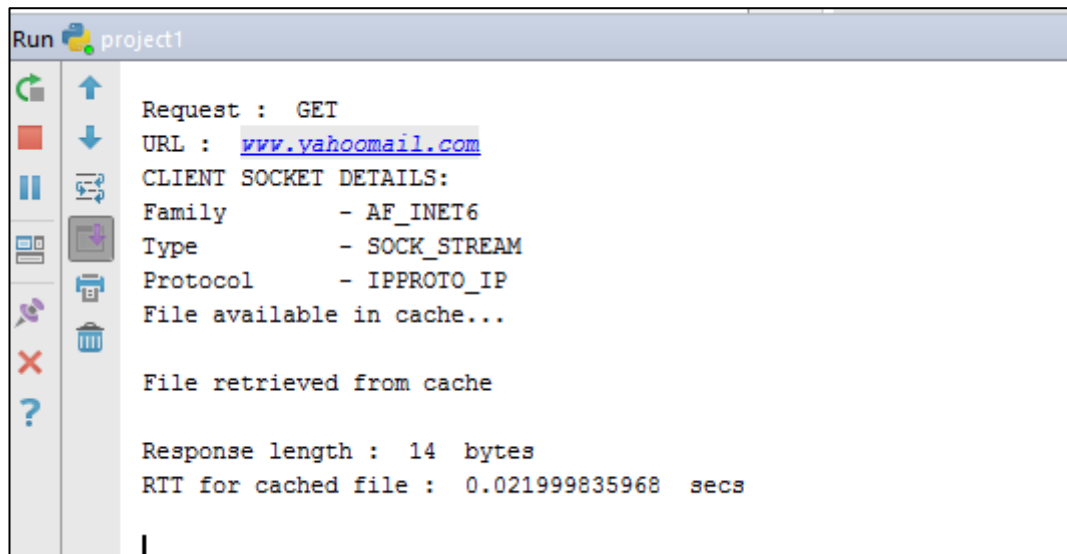
- Upon entering a site address, it checks for the available cache. If the cache is not available, the site is loaded from the web server and correspondingly a cache is made for future use.



```
Run project1
CLIENT SOCKET DETAILS:
Family      - AF_INET6
Type        - SOCK_STREAM
Protocol    - IPPROTO_IP
Could not find file in cache...
Retreiving file from server...
PROXY SERVER SOCKET DETAILS:
Family      - AF_INET6
Type        - SOCK_STREAM
Protocol    - IPPROTO_IP
Host Name   - www.yahooomail.com
Socket connected to host at port 80
Length of response from web server to proxy : 551 bytes
Caching Done!!!
RTT for file from web server : 1.53800010681 Waiting for requests...
```



- If the page was already loaded and it's a cache hit, the page is loaded from the local cache, and not from the real web server.



```

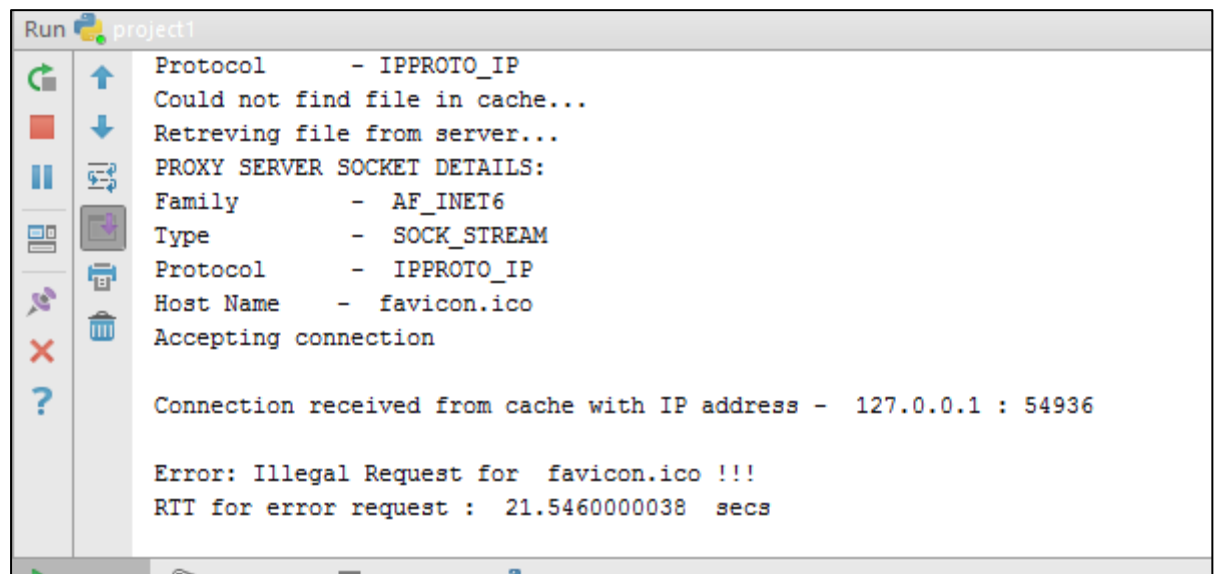
Run project1
Request : GET
URL : www.yahoo.com
CLIENT SOCKET DETAILS:
Family      - AF_INET6
Type        - SOCK_STREAM
Protocol    - IPPROTO_IP
File available in cache...

File retrieved from cache

Response length : 14 bytes
RTT for cached file : 0.021999835968 secs

```

- If an illegal request is made, it is marked as error and 404 error message is send to the client.



```

Run project1
Protocol    - IPPROTO_IP
Could not find file in cache...
Retreiving file from server...
PROXY SERVER SOCKET DETAILS:
Family      - AF_INET6
Type        - SOCK_STREAM
Protocol    - IPPROTO_IP
Host Name   - favicon.ico
Accepting connection

Connection received from cache with IP address - 127.0.0.1 : 54936

Error: Illegal Request for favicon.ico !!!
RTT for error request : 21.5460000038 secs

```



This localhost page can't be found

No webpage was found for the web address: <http://localhost:8080/test.gm>

Search Google for [localhost 8080 test](#)

HTTP ERROR 404

TASKS COMPLETED

1. All the details printed to the console as logged to the file log.txt
2. All the response and request messages are printed to console and logged to log file, along with its length
3. Multi-threading is incorporated to handle multiple requests at the same time
4. Caching of files are done
5. All the http messages exchanged by the proxy server must be automatically written into a file called log.txt.
6. The host address, local port, host name, request length (bytes sent), response length (bytes returned) and time elapsed (RTT), error message, etc. are logged.
7. Server details are printed on client side
8. Client details are printed on server side
9. Receive/send messages from/to a web browser
10. Receive/send messages from/to a web server
11. Process GET messages

Calculation of RTT

RTT is calculated and printed with each request.

RTT for cached files are less than the RTT for files retrieved from real web server as the data is locally available and hence access time is less.

Example:

In the case of URL: www.yahoomail.com

RTT for file from web server: 1.35800004005 secs

RTT for cached file: 0.0239999294281 secs

Referenced Websites:

- <https://docs.python.org/2/library/socket.html>
- http://www.tutorialspoint.com/python/python_multithreading.htm
- <https://wiki.python.org/moin/TcpCommunication>
- <https://pymotw.com/2/socket/tcp.html> (To get socket properties, code used)
- <https://docs.python.org/2/howto/sockets.html>
- https://www.tutorialspoint.com/python/python_networking.htm
- <https://docs.python.org/3/howto/logging.html>