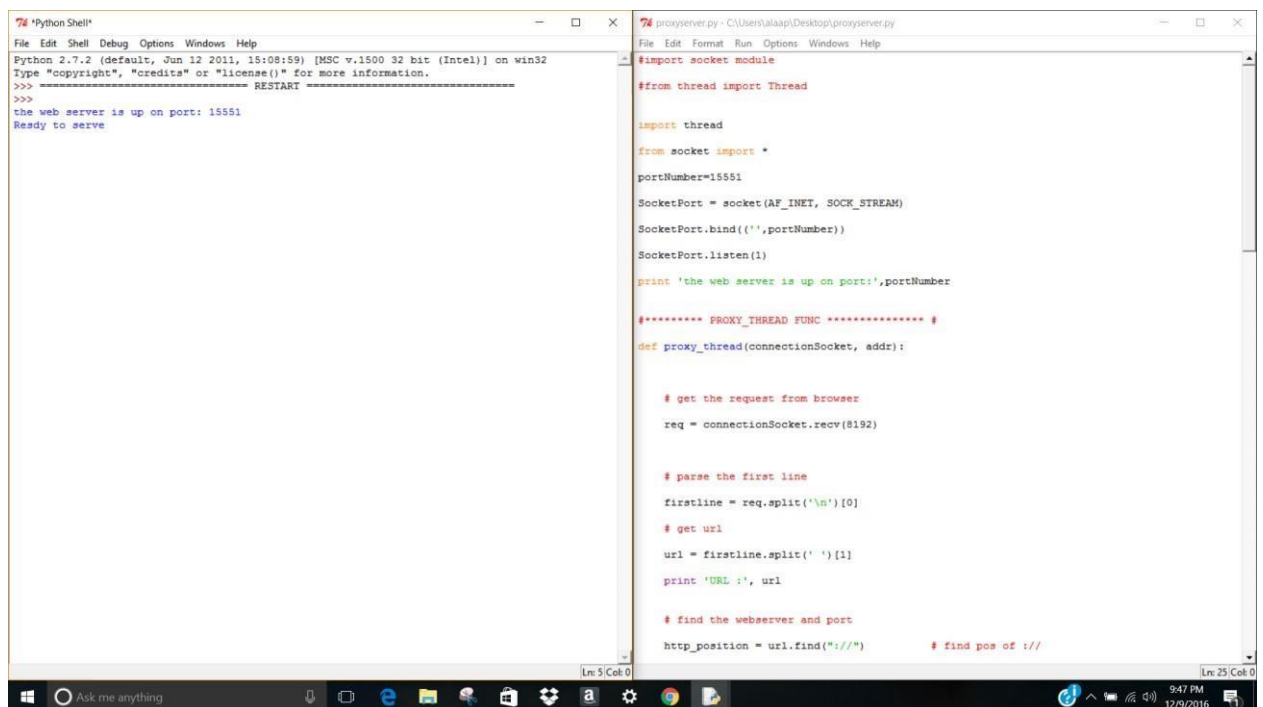


NA PROJECT 3B

ALAAP VASIREDDY

- If a client requests a Web page that is in the server cache, then the server will return the Web page to the client.
- If a client requests a Web page that is not in the server cache, then the server will open a connection to another server (either specified by the client or pre-specified) and return that Web page to the client.
- We made the local host as proxy server by changing the settings in the browser.



The image shows two side-by-side Python Shell windows. The left window is titled 'Python Shell' and shows the prompt 'Python 2.7.2 (default, Jun 12 2011, 18:08:59) [MSC v.1500 32 bit (Intel)] on win32'. It displays the command 'RESTART' and the output 'the web server is up on port: 15551' and 'Ready to serve'. The right window is titled 'proxyserver.py - C:\Users\alaap\Desktop\proxyserver.py' and shows the source code for a proxy server. The code imports the socket module and the Thread class from the threading module. It defines a port number 15551, creates a socket object, binds it to all interfaces on the specified port, and starts listening. A function 'proxy_thread' is defined to handle incoming requests by receiving data from the browser, parsing the first line to extract the URL, and then finding the webserver and port to proxy the request to. The code is currently at line 25, column 0.

```
File Edit Shell Debug Options Windows Help
Python 2.7.2 (default, Jun 12 2011, 18:08:59) [MSC v.1500 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> ===== RESTART =====
>>>
the web server is up on port: 15551
Ready to serve

File Edit Format Run Options Windows Help
proxyserver.py - C:\Users\alaap\Desktop\proxyserver.py
import socket module
from thread import Thread

import thread
from socket import *
portNumber=15551
SocketPort = socket(AF_INET, SOCK_STREAM)
SocketPort.bind(('',portNumber))
SocketPort.listen(1)
print 'the web server is up on port:',portNumber

#***** PROXY_THREAD FUNC *****#
def proxy_thread(connectionSocket, addr):

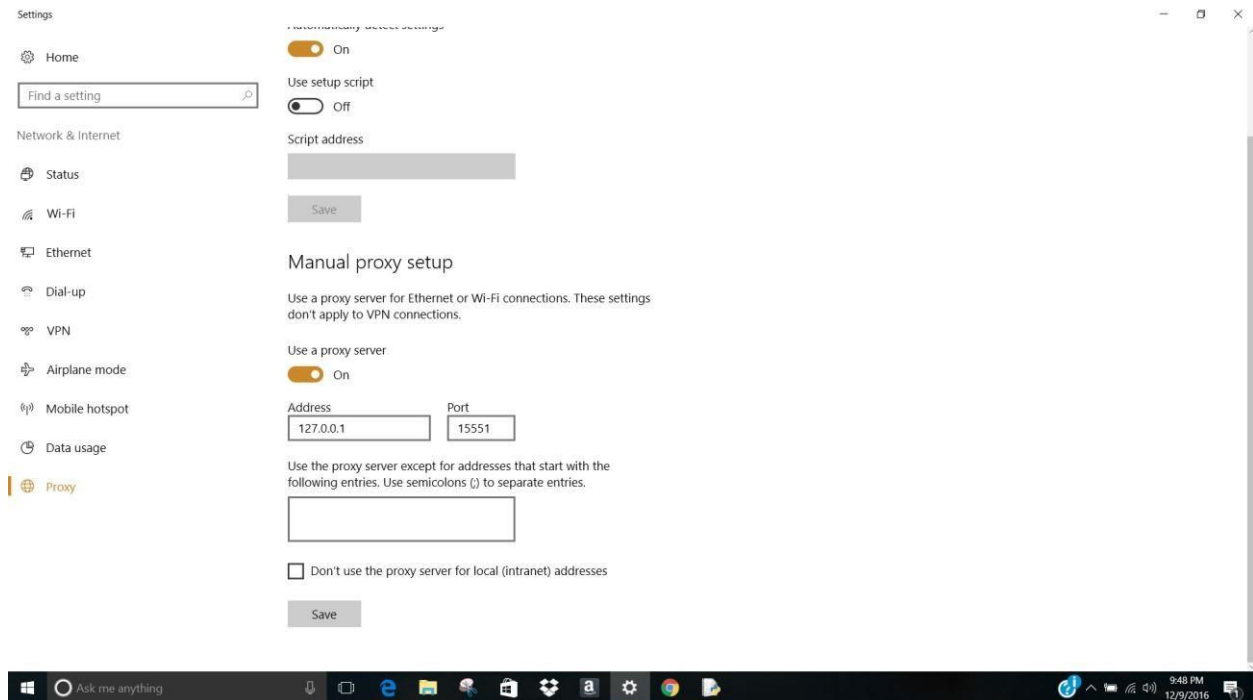
    # get the request from browser
    req = connectionSocket.recv(8192)

    # parse the first line
    firstline = req.split('\n')[0]

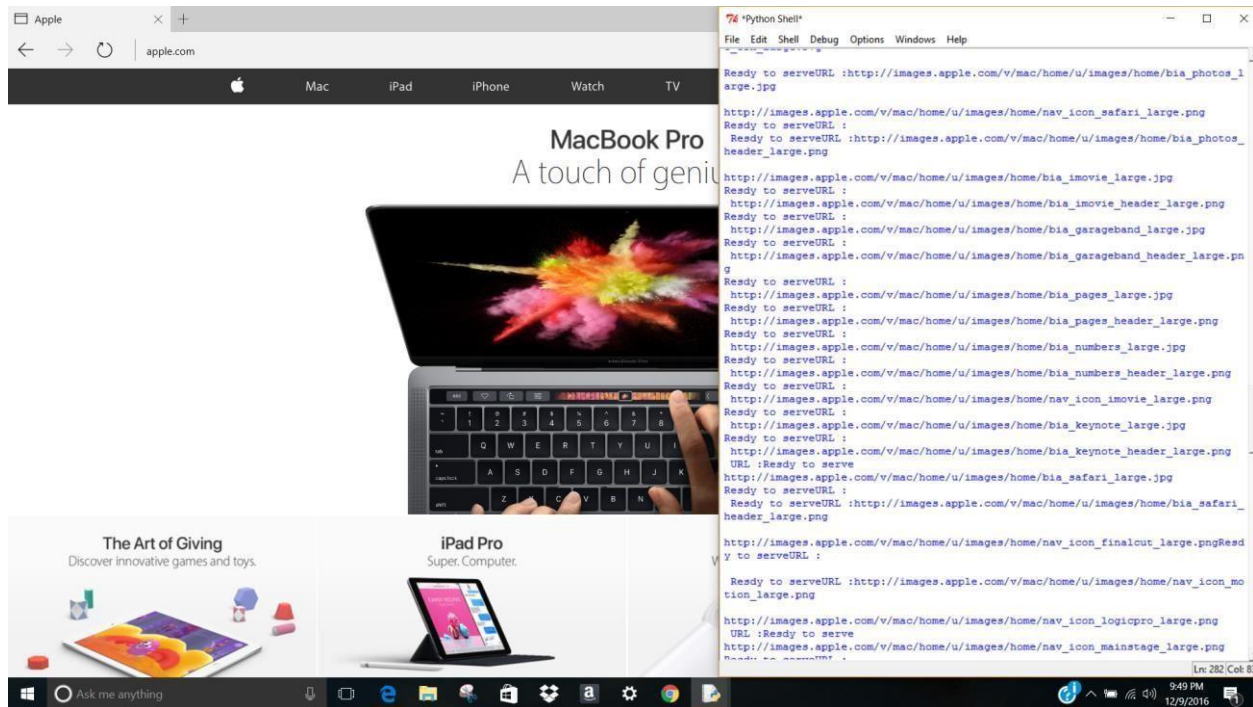
    # get url
    url = firstline.split(' ')[1]
    print 'URL :', url

    # find the webserver and port
    http_position = url.find("http://") # find pos of http
```

3b.1 Running the web server



3b.2 Setting the LocalHost



3b.3 Running the website using LocalHost

References

<http://luugiathuy.com/2011/03/simple-web-proxy-python/>

<http://null-byte.wonderhowto.com/how-to/sploit-makeproxy-server-python-0161232/>