

ASSIGNMENT 1

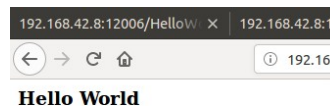
Q1. Write a Client Server program to simulate multi-clients single-server network.

SERVER.PY

```
#import socket module
from socket import *
serverSocket = socket(AF_INET, SOCK_STREAM)
#Prepare a sever socket
serverSocket.bind(('', 12006))
serverSocket.listen(1)
while True:
    print 'Ready to serve...'
    #Establish the connection
    connectionSocket, addr = serverSocket.accept()
    try:
        message = connectionSocket.recv(1024)
        filename = message.split()[1]
        f = open(filename[1:])
        outputdata = f.read()
        f.close()
        #Send one HTTP header line into socket
        connectionSocket.send('HTTP/1.0 200 OK\r\n\r\n')
        #Send the content of the requested file to the client
        for i in range(0, len(outputdata)):
            connectionSocket.send(outputdata[i])
        connectionSocket.close()
    except IOError:
        #Send response message for file not found
        connectionSocket.send('404 Not Found')
        #Close client socket
        connectionSocket.close()
serverSocket.close()
```

HTML FILE

```
<html>
<body>
<h3>Hello World</h3>
</body>
</html>
```



```
administrator@swlab-cse-8: ~/115CS0228/Assignment 1
administrator@swlab-cse-8:~/115CS0228/Assignment 1$ python webserver.py
Ready to serve...
Ready to serve...
Ready to serve...
Ready to serve...
Ready to serve...
Ready to serve...
Ready to serve...
Ready to serve...
Ready to serve...
Ready to serve...
```

Q2. Write a socket program to check if a particular port is open or closed.

```
import socket
sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
sock_name = socket.gethostbyname('www.nitrkl.ac.in')
result = sock.connect_ex((sock_name,80))
if result == 0:
    print "Port is open"
else:
    print "Port is not open"
```

```
administrator@swlab-cse-8: ~/115CS0228/Assignment 1
administrator@swlab-cse-8:~/115CS0228/Assignment 1$ python check_port.py
Port is open
administrator@swlab-cse-8:~/115CS0228/Assignment 1$
```