Server:

```
## import `socket` module
from _thread import *
import threading
from socket import *
import sys # In order to terminate the program
print_lock = threading.Lock()
serverSocket = socket(AF_INET, SOCK_STREAM)
# Prepare a sever socket
host = "127.0.0.1"
# Fill in start
port = 6789
serverSocket.bind((host,port))
serverSocket.listen(5)
# Fill in end
def threaded(connectionSocket):
while True:
# Establish the connection
print('ready to serve...')
#connectionSocket, addr = serverSocket.accept()

try:
message = connectionSocket.recv(1024)
filename = message.split()[1]
f = open(filename[1:])
outputdata = f.read()

# send an HTTP OK header line into socket
# Fill in start
connectionSocket.send(b'\nHTTP/1.1 200 OK\n\n')
## TODO: call the proper function with argument (b'\nHTTP/1.1 200 OK\n\n')
## Note that b'' converts the string into UTF-8-encoded bytes

#Fill in end

#Send the content of the requested file to the client
for i in range(0, len(outputdata)):
try:
connectionSocket.send(outputdata[i].encode())
except (ConnectionResetError, BrokenPipeError):
print("Connection closed unexpectedly by the client")
break
connectionSocket.send("\r\n".encode())


#connectionSocket.close()
break
```

```
except IOError:
# send response message for file not found
# fill in start
try:
connectionSocket.send(b"HTTP/1.1 404 Not Found\r\n\r\n")
connectionSocket.send(b"<html><head></head><body><h1>404 Not
Found</h1></body></html>\r\n")
except (ConnectionResetError, BrokenPipeError):
print("Connection closed unexpectedly by the client")
break #fill in end

#close client socket
#fill in start
#connectionSocket.close()
#fill in end
break
print_lock.release()
connectionSocket.close()

while True:
# establish connection with client
c, addr = serverSocket.accept()
# lock acquired by client
print_lock.acquire()
print('Connected to :', addr[0], ':', addr[1])
# Start a new thread and return its identifier
start_new_thread(threaded, (c,))



serverSocket.close()

sys.exit()
```

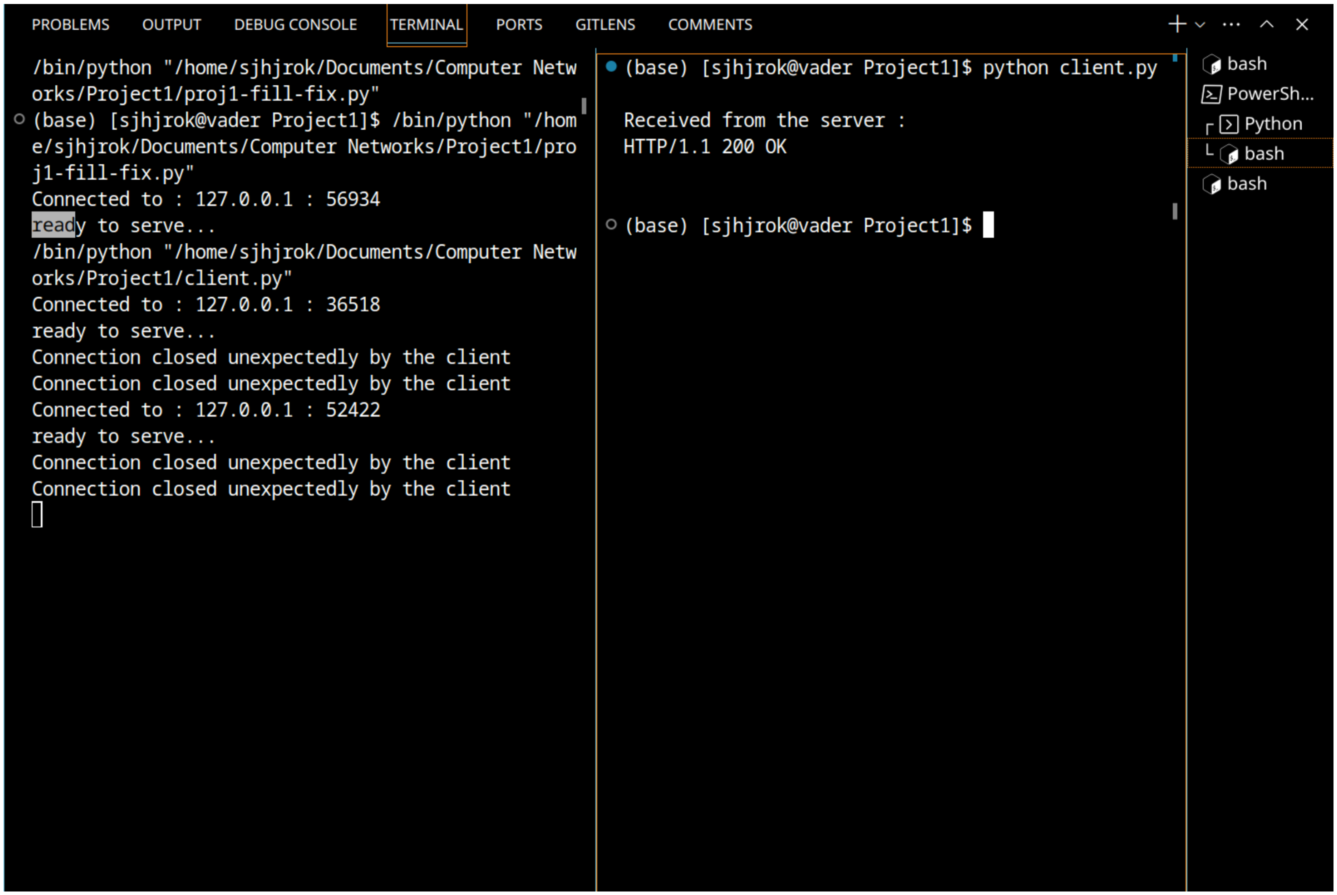
Client:

```
## import `socket` module
from _thread import *
import threading
import socket
import sys # In order to terminate the program
import argparse
def create_parser() -> argparse.ArgumentParser:
"""
Create an argument parser object using the `argparse` module in
Python.
Returns:
```

```
argparse.ArgumentParser: The argument parser object.
"""
parser = argparse.ArgumentParser(prog="client")
parser.add_argument("--address", type=str, default='127.0.0.1',
help="server address")
parser.add_argument("--port", type=int, default=6789, help="port
number")
parser.add_argument("--message", type=str, default="GET
/HelloWorld.html")

return parser
parser = create_parser()
args = parser.parse_args()
# local host IP '127.0.0.1'
host = args.address

# Define the port on which you want to connect
port = args.port

s = socket.socket(socket.AF_INET,socket.SOCK_STREAM)

# connect to server on local computer
s.connect((host,port))

# message you send to server
message = args.message
while True:
# message sent to server
s.send(message.encode('ascii'))

# message received from server
data = s.recv(1024)

# print the received message
# here it would be a reverse of sent message
print('Received from the server :',str(data.decode('ascii')))

break
# close the connection
s.close()
```

Client/Server:

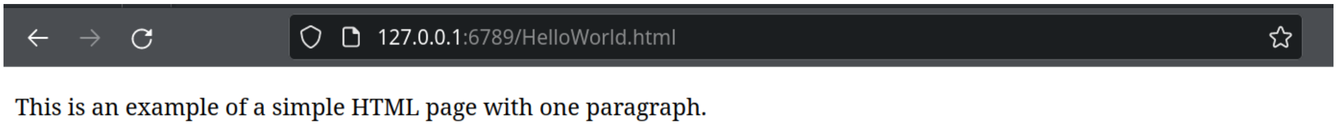
The client has parameter inputs set up using a parser however they have default settings that I just use to test and what I took a screenshot of.



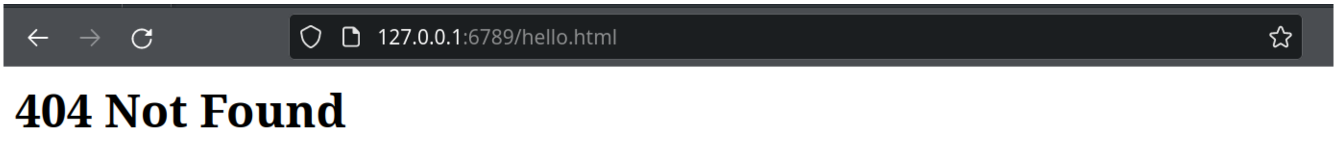
```
/bin/python "/home/sjhjrok/Documents/Computer Netw
orks/Project1/proj1-fill-fix.py"
○ (base) [sjhjrok@vader Project1]$ /bin/python "/hom
e/sjhjrok/Documents/Computer Networks/Project1/pro
j1-fill-fix.py"
Connected to : 127.0.0.1 : 56934
ready to serve...
/bin/python "/home/sjhjrok/Documents/Computer Netw
orks/Project1/client.py"
Connected to : 127.0.0.1 : 36518
ready to serve...
Connection closed unexpectedly by the client
Connection closed unexpectedly by the client
Connected to : 127.0.0.1 : 52422
ready to serve...
Connection closed unexpectedly by the client
Connection closed unexpectedly by the client
```

```
● (base) [sjhjrok@vader Project1]$ python client.py

 Received from the server :
 HTTP/1.1 200 OK


○ (base) [sjhjrok@vader Project1]$
```

Web Browser:



This is an example of a simple HTML page with one paragraph.



404 Not Found