Lab 1 - Web Server Report

Chad Gulczynski
CS4220
Professor Tan
2/18/26

Implementation Summary

This project creates a simple web server using Python's built-in server socket.

Key Functions and their Behaviors

[1] Python Socket Programming Guide

Preparing the Server Socket

```
serverSocket = socket(AF_INET, SOCK_STREAM)
```

- the `socket()` function instantiates our socket object for the lab which will use the necessary methods summarized below to establish the socket connection on the server side.
- the `AF_INET` argument specifies that the socket will use IPv4.
- the `SOCK_STREAM` argument specifies that the socket will use the TCP transport protocol for message transmission.

```
serverSocket.bind((SERVER_ADDR, SERVER_PORT))
```

- the `bind()` method links the socket address—*composed of a tuple containing the server's network address and the web server's port number*—to the socket object.

```
serverSocket.listen()
```

- the `listen()` method simply allows the web server to accept incoming connections to the socket.

Establishing the Connection

```
connectionSocket, clientAddr = serverSocket.accept()
```

- the `accept()` method accepts an incoming connection and returns:
  1. a new socket object `connectionSocket` that's used only for communication with the incoming client.
  2. the IP address `clientAddr` of the incoming client .

```
message = connectionSocket.recv(1024)
```

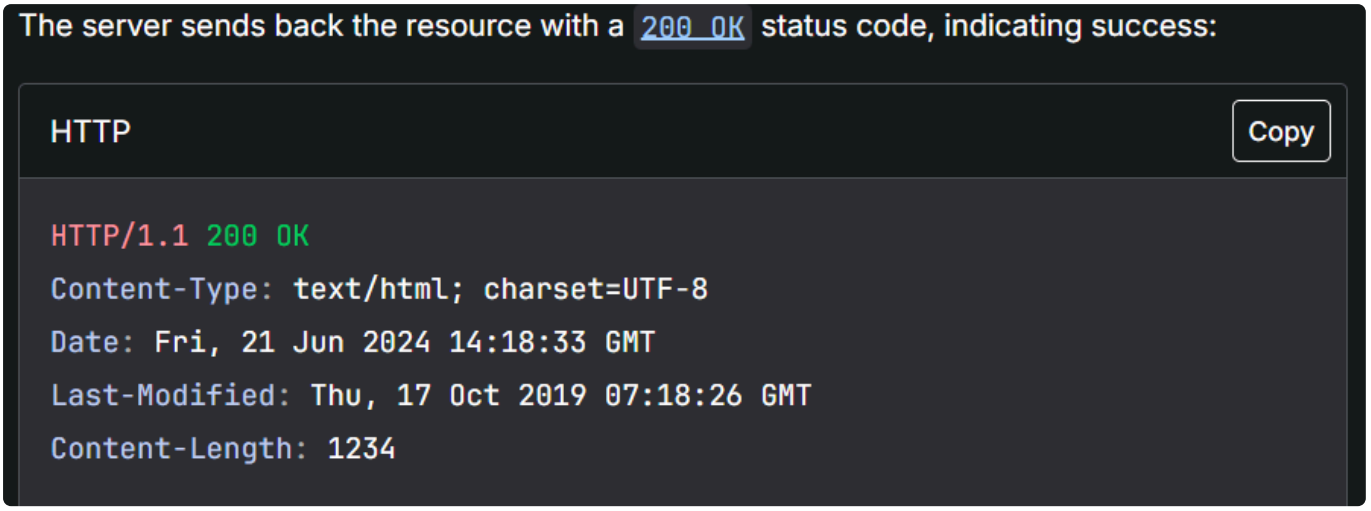
- the `recv()` method is used to receive data via our new socket connected to the client.
- the `1024` argument specifies the buffer size.

> **Note**
> I wasn't sure what value to pass for the buffer size argument, but the sources that I used consistently used 1024. From my understanding the value can be anything, but larger values can consume unnecessary memory and small values will require more calls to `recv()` to get the entire message. [2] Python Socket `recv`

```
outputdata = f.read()
```

- the `read()` method takes the `HelloWorld.html` file and reads the contents of the webpage into `outputdata`

Send One HTTP Header Line into Socket



[3] HTTP Response Header

```
connectionSocket.send(b"HTTP/1.1 200 OK\r\n")
connectionSocket.send(b"\r\n")
```

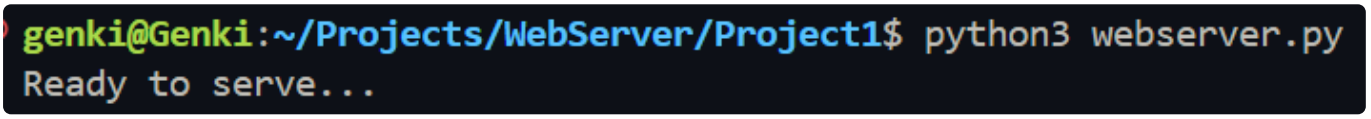
- Using the references above to craft a correct `GET` response, the `send()` method is used to send one HTTP header line into the socket.
  - HTTP message format [4] RFC 2616

Send Response Message for File not Found

```
connectionSocket.send(b"HTTP/1.1 404 Not Found\r\n")
```

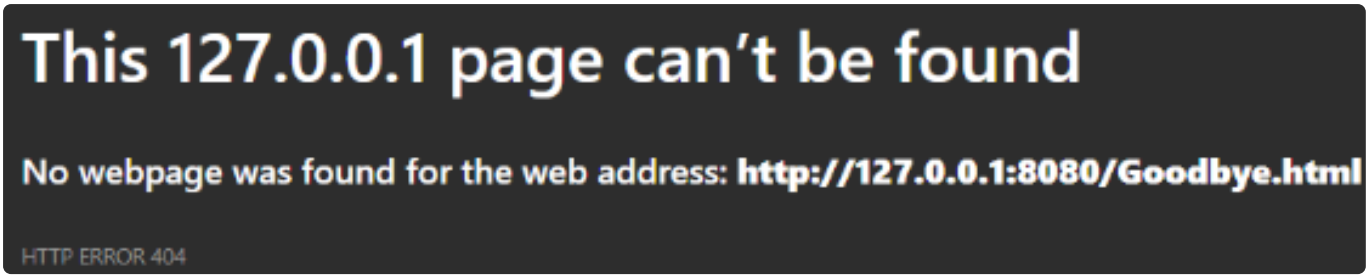
Validation Evidence

Running Web Server



Accessing `HelloWorld.html`

Accessing Non-Existing Page



Browser Network Output

Name	Status	Type
HelloWorld.html	200	document
Goodbye.html	404	document

Attribution

[1] Socket Programming in Python (Guide) – Real Python
[2] Python Socket `recv` - A Comprehensive Guide - CodeRivers
[3] GET request method - HTTP Response Header | MDN
[4] RFC 2616: Hypertext Transfer Protocol -- HTTP/1.1