

Python Lab 1: Web Server

Nitish Dhinakaran

University of the Cumberlands

MSCS-631-M40 Advanced Computer Networks

Dr. Dax Bradley

September 7, 2025

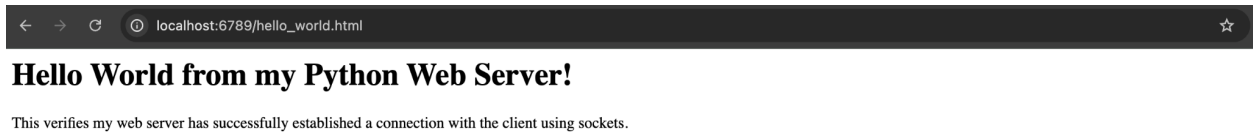
Web Server

Github Link

https://github.com/ndhinaharan36295/MSCS-631_Advanced-Computer-Networks/tree/main/Python-Lab-1_Web-Server

Output screenshot

My webserver was able to successfully establish a connection with the client on the specific port and displayed the HTML file.



Experience and Challenges

Working on this lab gave me hands-on experience with socket programming and a better understanding of how a basic web server functions. I was able to see how a server can be created, bound to a port, and used to listen for client requests. By completing the skeleton code, I gained practical insight into parsing HTTP requests, sending proper response headers, and returning the

requested HTML file to the browser. It was rewarding to load my `hello_world.html` file in the browser and know that the content was served directly from the Python program I built.

One of the challenges I faced was correctly handling the parsing of the request message to extract the filename, as well as ensuring the proper HTTP header format was sent back to the client. I overcame this by reviewing the structure of HTTP requests, breaking the request line into tokens, and carefully testing my code with different file names until it consistently returned the correct resource. Another difficulty was understanding how the server should respond when a file is not found, and formatting a “404 Not Found” response that browsers could interpret properly. To resolve this, I researched basic HTTP response codes and implemented a simple HTML message within the error response, which worked as expected in the browser.



Additionally, managing sockets so they closed correctly after each request required careful attention, since leaving sockets open caused unexpected behavior. I overcame this by ensuring that `connectionSocket.close()` was always called in both the normal and exception paths, which stabilized the server’s behavior. Despite these challenges, working through the debugging process helped me strengthen my understanding of the flow of client-server communication and improved my confidence in troubleshooting networking code.