**CSCE 560**

**Socket Programming Assignment 2**

**Web Server**

**Fall 18**

**Assigned: 22 Oct**

**Due: 14 Nov, 1400**

In this lab, you will learn the basics using socket programming to implement a Web server in Python. Specifically, you will

* Create a socket and bind it to a specific address and port
* Create an HTTP header
* Send and receive HTTP traffic
* Create custom HTML pages

You will develop a web server that handles one HTTP request at a time. Your web server should accept and parse the HTTP request, get the requested file (e.g., helloworld.html) from the server’s file system, create an HTTP response message consisting of the appropriate HTTP header lines followed by the requested file, and then send the response directly to the client. If the requested file is not present, the server must send one packet back to the client containing

* an HTTP header “404 Not Found”
* a custom 404 HTML page (creative examples can be found at https://designschool.canva.com/blog/404-page-design/)

**Code**

The skeleton code for the Web server is on the course file server. You are to complete the skeleton code. The placeholders where you need to fill in code are marked with **#Fill in start** and **#Fill in end**. Each placeholders may require one or more lines of code. Your server must generate and send the appropriate HTTP header as well as the contents of the file (or error page/message). Remember, HTML headers are not displayed in the browser window. For example,

If the file is found, your server sends the following:

* HTTP header containing: **HTTP/1.1 200 OK (Not displayed in window)**
* Contents of file (e.g., helloworld.html)

If the file is not found, your server sends the following:

* HTTP header containing: **HTTP/1.1 404 Not Found (Not displayed in window)**
* Contents of 404 error file (e.g., customerror.html)

**Running the Server**

Place an HTML file (e.g., helloworld.html) in the same directory as the server. Run the server program. Determine the IP address of the host running the server (e.g., 10.1.1.123). From another host (if possible), open a browser and provide the corresponding URL. For example:

<http://10.1.1.123:6789/helloworld.html>

‘helloworld.html’ is the name of the file you placed in the server directory. Note also the port number after the colon. You need to replace this port number with whatever port you used in the server code; in this example, I used port number 6789. The browser should then display the contents of helloworld.html. If you omit ":6789", the browser will assume port 80 and you will get the web page from the server only if a web server is listening on port 80. I’ve found some browsers will only display the HTML and not render the page correctly. If this occurs, try a different browser.

Then try to retrieve a file not found on the server. You should receive a “404 Not Found” HTTP header and the contents of your custom web page.

**What to Hand in**

You will hand in the following:

1. Hardcopy of complete server code
2. Screenshot of your client browser when you successfully retrieve a file from the server
3. Screenshot of Wireshark (filter out all packets except those pertaining to the HTTP transaction) when you successfully retrieve a file from the server
4. Screenshot of your client browser when the requested file is not found on the server
5. Screenshot of Wireshark (filter out all packets except those pertaining to the HTTP transaction) when the requested file is not found on the server
6. Source code of any transferred files (e.g., helloworld.html, customerror.html)