

Lab Project 4: Web Server using Python

- **Number of members per group: 1 or 2**
 - **Use Python**
-

In this lab, you will learn the basics of socket programming for TCP connections in **Python**: how to create a socket, bind it to a specific address and port, as well as send and receive a HTTP packet.

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 from the server's file system, create an HTTP response message consisting of the requested file preceded by header lines, and then send the response directly to the client. If the requested file is not present in the server, the server should send an HTTP "404 Not Found" message back to the client.

We assume that the client (web browser) can request **only html and jpg files**.

Additional Instructions:

Put an HTML file (e.g., HelloWorld.html) and a jpg file (scu_logo.jpg) in the same directory as the server program. Run the server program. Determine the IP address of the host that is running the server (e.g., 128.238.251.26). From another host, open a browser and provide the corresponding URL.

For example: `http://128.238.251.26:9000/html_file.html`

'html_file.html' is the name of the file you placed in the server directory. Note also the use of the port number after the colon. You need to replace this port number with whatever port you have used in the server code. In the above example, we have used the port number 9000. The browser should then display the contents of HelloWorld.html. If you omit ":9000", the browser will assume port 80 and you will get the web page from the server only if your server is listening at port 80.

To run the server, the user must provide port number as well in the following format:

```
python webserver.py 9000
```

Also:

- Try to get a file that is not present at the server. You should get a "404 Not Found" message.
 - To avoid "address already in use" error, use server option `SO_REUSEADDR`.
 - Your web server should be able to server both html and jpg files. **If no file is specified in the http request header, the server returns index.html file.**
 - The server also prints (in the terminal) the received http header as well as each prepared http header before sending to client.
-

Deliverables:

- **Demo your project to the TA in the lab**
- **Submit your code to Camino**