ENSF 310: Lab #6

A Simple Python HTTP Server

Due: Friday, November 26th

In this lab, you will start with the Simple Socket Server example from class.

Part 1: GET Requests

The provided `MyHTTPServer.py` file can be used a starting point. As a reminder, to run the server, simply run the `.py` file and then visit `http://localhost:12345/` in your web browser. Your task in this lab will be to implement the following functionality:

- Visiting `http://localhost:12345/test.html` will read the `test.html` file and send it to the browser.
- Visiting `http://localhost:12345/` will, at minimum, show 'Welcome to my Python Server' in the browser.
- Visiting `http://localhost:12345/?name=Roman` will, at minimum, show '{name:Roman}' or something similar in the browser.
- Visiting `http://localhost:12345/?name=Roman&class=ENSF310` will, at minimum, show '{name:Roman,class:ENSF310}' in the browser.

The above are minimum requirements and you may be more creative in what you show in the browser.

Part 2: POST Requests

Now, extend your Python HTTP Server to also handle POST requests. You should first create an html file which would result in a POST request (likely using a web form of some type). Your server should simply show the variables sent via POST in the browser, similar to the behavior of the GET request with variables being passed via a `?`.

Resources:

- GET vs POST requests: https://www.w3schools.com/tags/ref_httpmethods.asp (https://www.w3schools.com/tags/ref_httpmethods.asp)
- Form Data: https://developer.mozilla.org/en-US/docs/Learn/Forms/Sending_and_retrieving_form_data (https://developer.mozilla.org/en-US/docs/Learn/Forms/Sending_and_retrieving_form_data)