William Frazier

My server always expects a message to end with “\n", if the newline character is missing then it won't process the message. This decision was made so that the server could know when the full message was received (I was having trouble using the system I used in part 1). I initially included a timeout to ensure the server wouldn’t keep waiting for a newline that was never coming but I kept hitting weird edge cases and eventually removed it. Also, the server throws a 404 if the message payload contains a space. This was to aid in checking that white space is correct. I figured that the ease of coding a server with these restrictions was worth the tradeoff of not allowing certain messages.

I built the server to allow a queue of 5 connections (following a recommendation that I found online). I also added a really ugly error checking function because it allowed me a nice way to print out the exact error encountered. Other than that, there’s really not too much to say about my programs, they basically just follow the design specifications. I also want to make sure I note that much of my code from part 1 is code available online because I used some online guides to see how to code sockets in python. All added code in part 2 is my own.