A screen capture is included in the zip file under the name “/content/videos/demo.mp4”.

**Basic logic:**

The program begins listening on the server socket and creates a connection socket in a new thread that receives each http request from the browser. It then parses the request to check if the file exists while also checking what type of file and if the server should stop running. If the file is found it creates the header and the content and then uses the sendMessage() function I wrote to send the data back to the web browser.

If the file does not exist or if the content type does not exist it creates a 404 response and responds with the 404 html page.

If the last four characters of the requested file are “stop” then the server will quit the for loop, send one last page to let the user know the server is done running, and then close the server socket.

I implemented the program in Python 3 with the os and socket libraries.

**Problems encountered:**

I had a lot of difficulty getting the http response formatted correctly, I thought the “\r\n\r\n” sequence was needed after the content rather than before so I spent some time trying to figure out if I had the wrong sequence, location, or if the html file was the problem.

I wanted the server to be able to be stopped from the browser somehow so I was able to use the parsing and added some modifications so that a path ending with “stop” (no file type) will cause the program to gracefully exit.

I originally wrote it as a single thread program, and changed it using \_thread which was a bit confusing but I got it working. I then heard about the multi-threading example in class and switched it over to that easily.