Sunny Ka Patel
SDK438
11267665
CMPT 353.A2.Part-C

- I did my testing iteratively where in each iteration I tested some functionality and build confidence upon each successful test. First, I tested whether clicking the button sent a POST request to the server or not. After successful testing of that, I tested if the posts.txt file was created and written with the dummy inputs I provided or not. Upon successful testing, I tested if the dummy inputs are displayed on the browser or not. Finally, for integration testing, I inputted couple of dummy values and checked whether each post is displayed right after it is created or not.
- When I increased the length of the post, at around 100000 paragraphs I got a message on my terminal saying PayloadTooLargeError: request entity too large. Prior to that I noticed little to no effect to the performance.
- Using *loadtest*, I got the following results:

n	**c**	**Time taken**
1	100	0.018289613 s
10	100	0.028257382 s
100	100	0.064367411 s
1000	100	0.40288653 s
10000	100	2.913417188 s

n	**c**	**Time taken**
100	10	0.078529411 s
100	100	0.065752729 s
100	1000	0.079980811 s
100	10000	0.104594652 s

- From the above results, for just a single request, the time taken is ~ 0.0183 seconds.
- When the level of concurrency is kept the same while the number of requests is increased, the time taken to complete all request increased, i.e., decreased performance.
- When the number of requests is kept the same while the level of concurrency is increased, we can see an increase in performance (i.e., less time taken) but up to a certain limit.