Socket Programming Project – Python HTTP Server, *Parallel Edition*

Name	Major	Primary Objective (60 pts)	Secondary Objectives (40 pts)	Total Score (100 pts)

Evaluation Rubric – Primary Objective (60 pts possible):
Web server can support concurrent requests from multiple clients simultaneously

Description	Points	Score
Server successfully uses one of the following methods to service multiple client sockets concurrently: threads, processes, select(), or event-driven API. Readme.txt file briefly describes parallelism choice.	**45**	
FunkLoad benchmark configuration and results submitted for Project 1 server and Project 2 server. Benchmark was *not* done to localhost.	**15**	
Total (60 pts possible)		

Evaluation Rubric – Secondary Objectives (40 pts possible):

Description	Points	Score
Server supports HTTP persistent connections (socket stays open 30 seconds after a request, waiting for more requests)	**10**	
Server supports HTTP pipelined connections (multiple requests can be sent back-to-back without waiting for server first)	**10**	
Server gracefully shuts down upon receiving user CTRL-C	**5**	
Server supports HTTP HEAD requests	**5**	
Server provides the following HTTP response headers: Date, Server, Content-Length, Content-Type, Last-Modified, Expires	**5**	
Server provides verbose and silent modes of operation	**5**	
Total (40 pts possible)		

Comments: