Socket Programming Project – Python HTTP Server

Name	**Major**	**Checkpoints** (50 pts)	**Primary Objective** (60 pts)	**Secondary Objectives** (40 pts)	**Total Score** (150 pts)

Checkpoints (50 points possible):
Checkpoints are graded as full credit *(25 points),* half credit *(13 points), or* no credit *(0 points)*

Description	**Points**	**Score**
Checkpoint 1: Server accepts request from client and displays request to screen (or, server does additional HTTP decoding). Server uses `argparse` for command-line arguments, including port number.	**25**	
Checkpoint 2: Server can successfully decode HTTP request from client and send a single reply file given a controlled test environment	**25**	
Total (50 pts possible)		

Evaluation Rubric – Primary Objective (60 pts possible):
Website loaded from project server is indistinguishable from site loaded from real server

Description	**Points**	**Score**
Server initializes socket subsystem and receives incoming HTTP requests on port 8080	**15**	
Server decodes incoming HTTP 1.1 requests	**15**	
Server sends requested file via HTTP 1.1	**15**	
Web browsers (Chrome, Firefox, Safari, IE) display website from Python HTTP server that is visually identical to site on “real” server	**15**	
Total (60 pts possible)		

Evaluation Rubric – Secondary Objectives (40 pts possible):

Description	**Points**	**Score**
Server supports required `--help`, `--base`, and `--port` command-line arguments and uses `argparse` to decode them	**10**	
Server supports HTTP response code 404	**10**	
Server runs in Python 3 (3.4.x or newer) in Linux environment *(should be free points here!)*	**10**	
Server submitted to Canvas site as a .tar.gz archive or .py file *(should be free points here!)*	**10**	
Total (40 pts possible)		

Comments: