

1. How did you decide which type of socket to use? Why?

I decided on using a TCP socket because my socket is sending and receiving HTTP packets. When transferring web objects, we want reliable transport and retransmission of lost packets. TCP gives these two aspects in its service. I used `socket.SOCK_STREAM` to indicate when creating the socket that it is a TCP socket.

2. How did you choose the destination ports?

My program parses the destination port from the user-given URL by matching the number after the colon symbol. If the port number is not given by the user, it will default to port 80, which is the port commonly used by web servers to listen on.

3. What error handling cases did you implement?

- HTTPS URLs: When parsing the URL scheme, check if HTTPS. If so, print an error message stating 'HTTPS is not supported.'
- Hosts that don't exist or can't be resolved: Catch `socket.gaierror` raised by address-related errors.
- Destination port is not 80 or 443: Set a timeout of 5 seconds for the socket. If socket hangs for more than 5 seconds, exit with Socket Exception error.
- Connection Reset/Connection Refused: Catch exceptions when sending or receiving bytes through socket.
- User only provides IP address in fullURL without a second argument specifying domain name: Check arguments length if host is an IP address, exit with error msg if second argument is not there.
- Empty reply from server: If 0 bytes are returned from HTTP response, print an error msg stating 'Empty reply from server'.
- Chunk encoding: Check if 'Transfer-Encoding' header has the value 'chunked'. If so, write to Log.csv and print an error message.
- Requested web object not returned: Check if status code is 200; if not, write to Log.csv and print 'Unsuccessful' message to terminal.

4. How does your program terminate? What happens to the TCP connection?

When an error occurs, my program closes the HTML and CSV file, and closes the socket before terminating. The TCP connection between client and server is also closed with `'client.socket.close()'`.

5. For the unsuccessful URLs, why were they unsuccessful?

- HTTPS in scheme part of URL
- Domain name could not be resolved
- Missing hostname if first argument is an IP address
- Requested web object could not be found/accessed

- Web object was returned with chunk encoding

6. What happens if you try to access a site using HTTPS?

My program terminates with an error message because the project specification states that URLs with HTTPS scheme should be rejected.