

# Questions

---

1. When initializing the socket, I decided to use `SOCK_STREAM`. This is because HTTP uses TCP, and `SOCK_STREAM` indicates that the socket is a TCP socket.
2. If no destination port was specified, I chose port 80, because that is the port used for the HTTP protocol. Otherwise, I grabbed the destination port from the URL in the command line argument.
3. I implemented error handling cases for parsing the URL and the hostname (checking for https, checking that a hostname or IP address is specified, checking that the port is valid, etc.). I also implemented error handling for the socket connection, for sending the request, getting and reading the header, and receiving the response. I then implemented corner cases like chunk encoding handling and timeouts.
4. My program terminates by closing the socket using `close()` and exiting the program. `close()` closes the socket and the TCP connection between the client and the server. The program can terminate on error, on success/failure, or if there is a timeout.
5. There are a number of reasons why the unsuccessful URLs were unsuccessful, including: the URL has https in it, the URL does not start with http, the IP address is invalid, the port is invalid, the domain name is invalid, the URL was relocated, etc.
6. If, using my program, we decide to access a site using HTTPS, we would not be able to because the program specification does not allow HTTPS. However, if my program did not check for HTTPS in the URL, the request would still be rejected, since HTTPS uses port 443 rather than port 80. Port 443 requires an SSL certificate, which is not supported by the program.