Advanced Computer Networks LAB
Spring Semester 2026
Assignment 1

Name – Soham Chakraborty
Roll No – 22CS02002

Part 1 –

1. **Problem:** Using web-browser as a client instead of a client.c code.
   Solution: Everything from the server-side remains the same. We have to just use *http://localhost:port_no* on the browser to act as client instead of a client.c.

2. **Problem:** Making a formatted webpage instead of printing output on terminal.
   Solution: I had to learn basic HTML enough to make a simple webpage.

Other than that, no such problems faced in this part due to previous CN Lab experience.

In this case, it was just a diminished version of a full-fledged continued client-server connection. Having previously done the same in C, with the help of previous CN Lab Project (6th Semester) it was doable.

Part 2 –

1. **Problem**: Unable to find OpenSSL headers during compilation.
   Solution: sudo apt install libssl-dev

2. **Problem**: Even after previous step and headers being included, functions from libraries still not working.

   Solution: Linker error. Using flags -lssl and -lcrypto to link.

3. **Problem:** Redirection issue when using HOST as “duckduckgo.com” [302 Moved Temporarily].

   Solution: Using HOST as "html.duckduckgo.com" instead.

4. **Problem**: Hit a captcha verification page [202 Accepted].

   Solution: Switching from **Institute LAN** to **Mobile Hospot (Airtel)**.
   Alternatively, change **User-Agent** from **Mozilla** to **Chrome.**

5. **Problem**: Incomplete/Partial output in SSL_read().

   Solution: Increasing **MAX_BUFFER_LENGTH** to 1024 (1 KB).

6. **Problem**: Error while using bind() function.

   Solution: Choosing a different port_no and host server on that one. Happened because the previous server was not closed properly, thus that port was still occupied from previous use.

Part 3 –

1. **Problem:** Incorrect received data on server side.

   Solution: This was happening because I didn’t clear out previous buffer character array. So, the information from previous receive from client was still present and it was merged into the new client request. Using *memset(buffer, 0, sizeof(buffer))* solved it.

Part 4 –

1. **Problem:** Making different URL do different tasks.

   Solution: Separating the path from the GET request sent by the browser/client and comparing the required commands or regex from it using string manipulation.

Part 5 –

No issues were faced in this part because we just had to design a calculator and it was the same as Part 4 where we had to string parse the URL and find the components from it.

Part 6 –

1. **Problem:** Using WebSockets instead of HTTP.

   Solution: Assignment Tutorial slides along with the help of ChatGPT to understand the basics of WebSockets. Such as WebSocket handshake and encoding and decoding messages while using send and receive

2. **Problem:** □WebSocket Protocol Error

   Solution: Unable to send payloads of size > 125 bytes. This was fixed by using a header to mention the size of payload first and then sending the payload in the next send. This helps to handle upto 65535 bytes and a separate case to handle more than that.

3. **Problem:** Unable to create Textbox and buttons on the webpage.

   Solution: I had to learn very basic JavaScript with the help of ChatGPT, since I have no previous knowledge about WebDevelopment or JavaScript.