NATIONAL INSTITUTE OF TECHNOLOGY SILCHAR

Cachar, Assam

B.Tech. Vth Sem

Subject Code: CS-311

Subject Name: Computer Network Laboratory

Submitted By:

Name : Subhojit Ghimire

Sch. Id. : 1912160

Branch : CSE – B

Q08. Write a program to implement "Web Server". (Description: The Client will be requesting a web page to be accessed which resides at the Server side.)

AIM: TO IMPLEMENT "WEB SERVER" USING TCP SOCKET IN CPP.

THEORY: 1. Web Server: A web server is computer software and underlying hardware that accepts requests via HTTP, the network protocol created to distribute web content, or its secure variant HTTPS.

2. TCP CLIENT SERVER: TCP (Transmission Control Protocol) is a transport layer in a networking service. The client in TCP/IP connection is the device that dials the phone and the server is the device that is listening in for the calls to come in.

CODE:

// WS SERVER

```
# include <iostream>
# include <cstdlib>
# include <cstring>
# include <netdb.h>
# include <netinet/in.h>
# include <sys/socket.h>
# include <sys/types.h>
# include <unistd.h>

# define MAX 1024
# define PORT 8080
using namespace std;

int main() {
    int sockfd = socket (AF_INET, SOCK_STREAM, 0);
    struct sockaddr_in servaddr;
    if ( sockfd != -1 )
        cout << " SUCCESS: SOCKET CREATED\n";
    else {
        perror ("ERROR: SOCKET CREATION ");
        exit (EXIT_FAILURE);
    }
    bzero (&servaddr, sizeof (servaddr));

    servaddr.sin_family = AF_INET;
    servaddr.sin_addr.s_addr = htonl (INADDR_ANY);
    servaddr.sin_port = htons (PORT);
```

```
if (!bind(sockfd, (struct sockaddr *)&servaddr, sizeof(servaddr)))
        cout << "SUCCESS: SOCKET BINDED\n";
else {
    perror ("ERROR : SOCKET BINDING");
    exit (EXIT_FAILURE);
}

struct sockaddr_in cli;
unsigned int len = sizeof (cli);
if (!(listen (sockfd, 5)))
        cout << "SUCCESS : SERVER LISTENING\n";
else {
    perror ("ERROR : SERVER LISTEN");
    exit (EXIT_FAILURE);
}

int connfd = accept (sockfd, (struct sockaddr *)&cli, &len);
if (connfd >= 0)
        cout << "SUCCESS : CLIENT ACCEPTED\n\n";
else {
        perror ("ERROR : CLIENT ACCEPTION");
        exit (EXIT_FAILURE);
}

char buffer[MAX], buff[MAX];
while (1) {
        bzero (buffer, MAX);
        bzero (buff, MAX);
        read (connfd, buffer, sizeof (buffer));
```

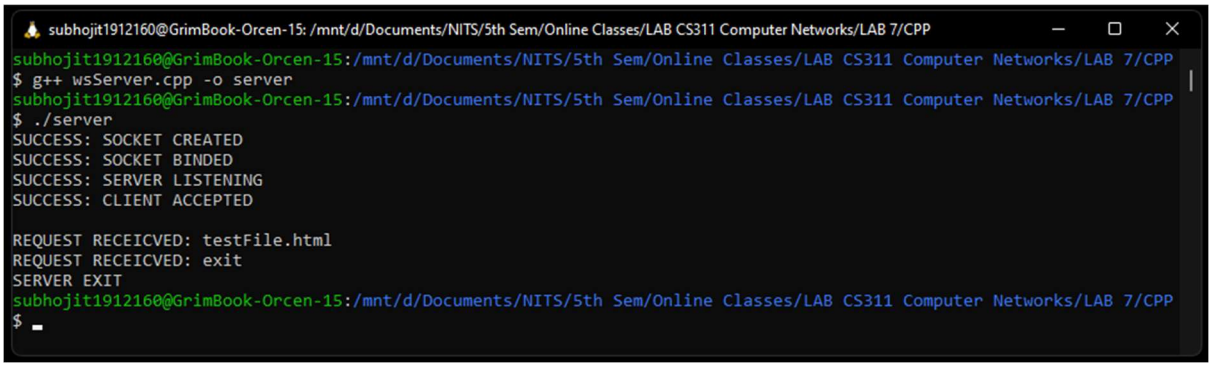
```
        strcpy (buff, buffer);
        buff [strlen (buff)-1] = '\0';
        cout << "REQUEST RECEIVED: " << buff <<endl;
        if (!strncmp (buff, "exit", 4))
                break;
        FILE *ff = fopen (buff, "r");
        while (!feof (ff)) {
                bzero (buffer, MAX);
                fgets (buffer, MAX, ff);
                write (connfd, buffer, sizeof (buffer));
        }
        fclose (ff);
        bzero (buffer, MAX);
        strcpy (buffer, "END");
        write (connfd, buffer, sizeof (buffer));
    }
    cout << "SERVER EXIT \n";
    close (sockfd);
    return 0;
}

// WS CLIENT

#include <iostream>
#include <cstdlib>
#include <cstring>
#include <netdb.h>
#include <arpa/inet.h>
#include <sys/socket.h>
#include <unistd.h>
```

```
#define MAX 1024
# define PORT 8080

int main () {
    int sockfd = socket (AF_INET, SOCK_STREAM, 0);
    struct sockaddr_in servaddr;
    if (sockfd != -1)
        cout << " SUCCESS: SOCKET CREATED\n";
    else {
        perror (" ERROR : SOCKET CREATION ");
        exit (EXIT-FAILURE);
    }
    bzero (&servaddr, sizeof (servaddr));

    servaddr.sin-family = AF_INET;
    servaddr.sin-addr.s-addr = htonl (INADDR_ANY);
    servaddr.sin-port = htons ( PORT);

    if (!(connect (sockfd, (struct sockaddr *) &servaddr,
        sizeof (servaddr))))
        cout << " SUCCESS : CONNECTE TO SERVER \n\n";
    else {
        perror (" ERROR: SERVER CONNECTION ");
        exit (EXIT_FAILURE);
    }

    cout << " ENTER FILENAME TO OPEN: ";
    char buffer [MAX], buff [MAX];
```

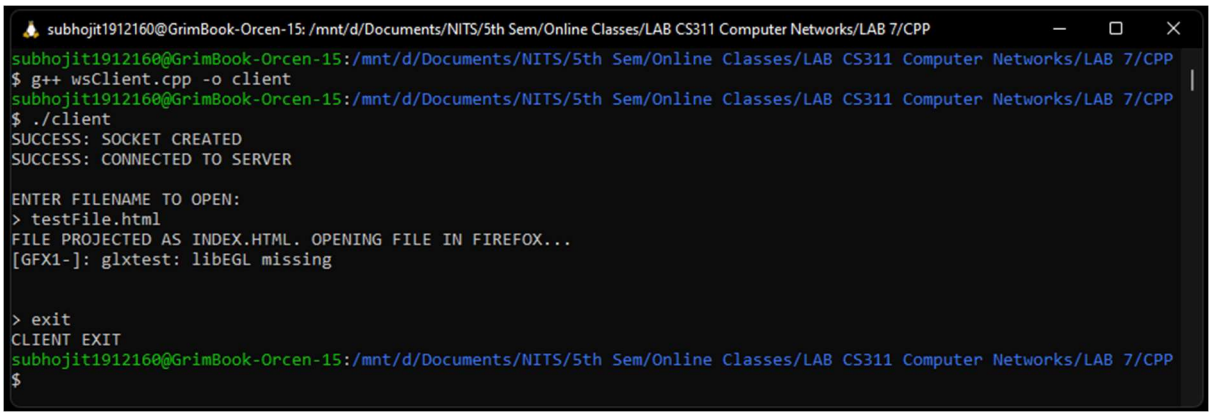
```
while (1) {
    bzero (buff, sizeof (buff));
    bzero (buffer, sizeof (buffer));
    cout << "\n> ";
    fgets (buff, MAX, stdin);
    write (sockfd, buff, sizeof (buff));
    if (!strncmp (buff, "exit", 4))
        break;
    FILE *ff = fopen ("index.html", "w");
    while (1) {
        bzero (buffer, sizeof (buffer));
        read (sockfd, buffer, sizeof (buffer));
        if (strncmp (buffer, "END", 3) == 0)
            break;
        fputs (buffer, ff);
    }
    fclose (ff);
    cout << "FILE PROJECTED AS INDEX.HTML.
            OPENING FILE IN FIREFOX...\n";
    system ("firefox index.html");
    int c = getchar ();
    system ("rm index.html");
}
cout << "CLIENT EXIT \n";
close (sockfd);
return 0;
}
```

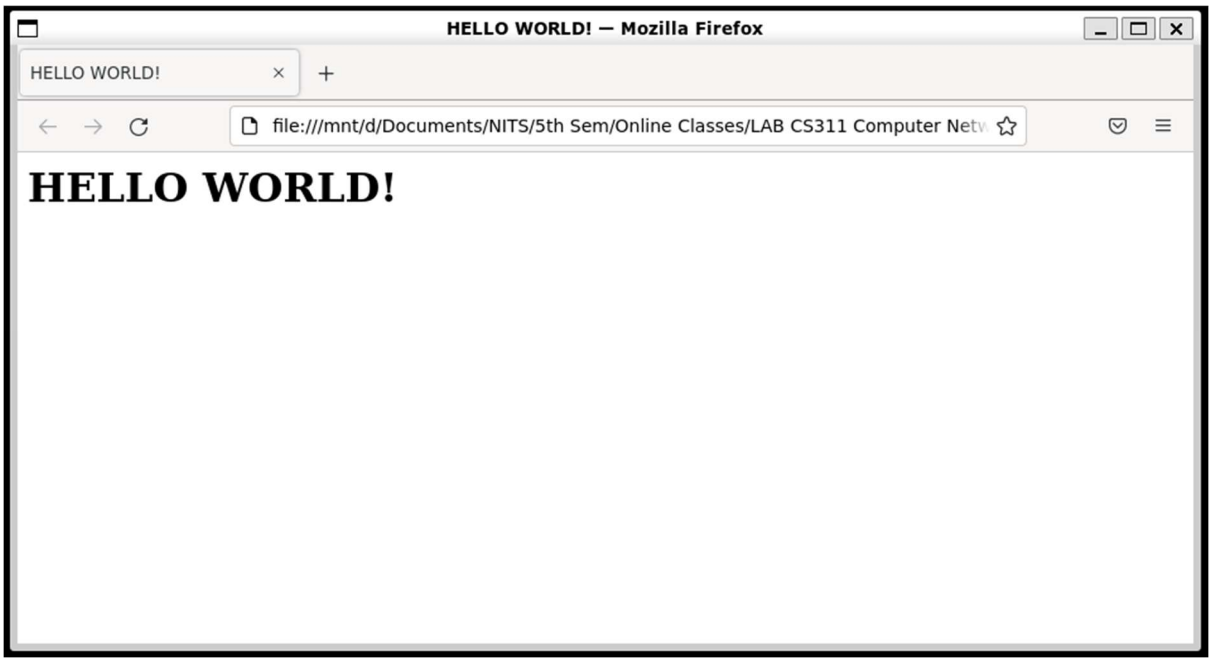
OUTPUT:

// WS SERVER



// WS CLIENT





Output Explanation:

The client requests a web-page from the server. The server receives the requests and responds accordingly by sending back the web-page resources so that the client can view the web-page on his/her side. The client program, after receiving the web-page, opens it on Firefox for the client to view. When the client is done, the “exit” command is sent to close the processes.