Experiment 3:

Client Code :

```c
#include <arpa/inet.h>
#include <netdb.h>
#include <netinet/in.h>
#include <stdio.h>
#include <string.h>
#include <sys/socket.h>
#include <sys/types.h>
#include <unistd.h>
int main(int argc, char *argv[])
{
    int cd, sd, ad;
    char buff[1024];
    struct sockaddr_in cliaddr, servaddr;
    struct hostent *h;
    h = gethostbyname(argv[1]);
    bzero(&servaddr, sizeof(servaddr));
    servaddr.sin_family = AF_INET;
    memcpy((char *)&servaddr.sin_addr.s_addr, h->h_addr_list[0], h->h_length);
    servaddr.sin_port = htons(2002);
    sd = socket(AF_INET, SOCK_STREAM, 0);
    cd = connect(sd, (struct sockaddr *)&servaddr, sizeof(servaddr));
    while (1)
    {
        printf("Enter the message: \n");
        fgets(buff, 100, stdin);
        send(sd, buff, sizeof(buff) + 1, 0);
        printf("\n Data Sent ");
        printf("%s", buff);
    }
    return 0;
}
```

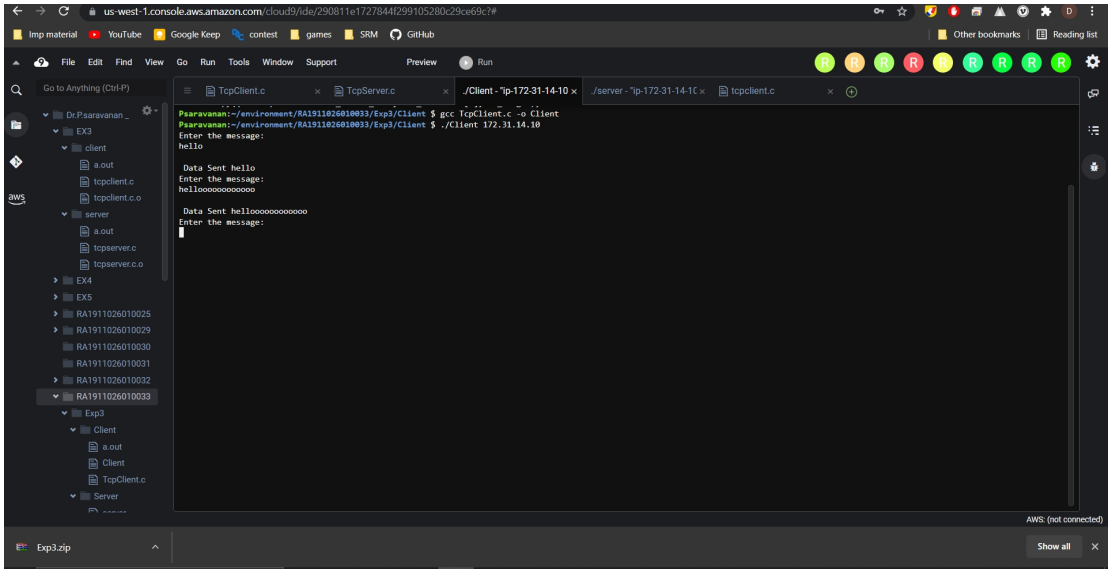
Server Code :

```c
#include <arpa/inet.h>
#include <netdb.h>
#include <netinet/in.h>
#include <stdio.h>
#include <string.h>
#include <sys/socket.h>
#include <sys/types.h>
#include <unistd.h>
int main(int argc, char *argv[])
{
    int bd, sd, ad;
    char buff[1024];
    struct sockaddr_in cliaddr, servaddr;
    socklen_t clilen;
    clilen = sizeof(cliaddr);
    bzero(&servaddr, sizeof(servaddr));

    servaddr.sin_family = AF_INET;
    servaddr.sin_addr.s_addr = htonl(INADDR_ANY);
    servaddr.sin_port = htons(2002);
    sd = socket(AF_INET, SOCK_STREAM, 0);
    bd = bind(sd, (struct sockaddr *)&servaddr, sizeof(servaddr));
    listen(sd, 5);
    printf("server is running");
    ad = accept(sd, (struct sockaddr *)&cliaddr, &clilen);
    while (1)
    {
        bzero(&buff, sizeof(buff));
        recv(ad, buff, sizeof(buff), 0);
        printf("The client message is: %s \n", buff);
    }
}
```

Output :

Client Output:



Server Output :