

21BCE1796

B SHAKTHI

# COMPUTER NETWORKS

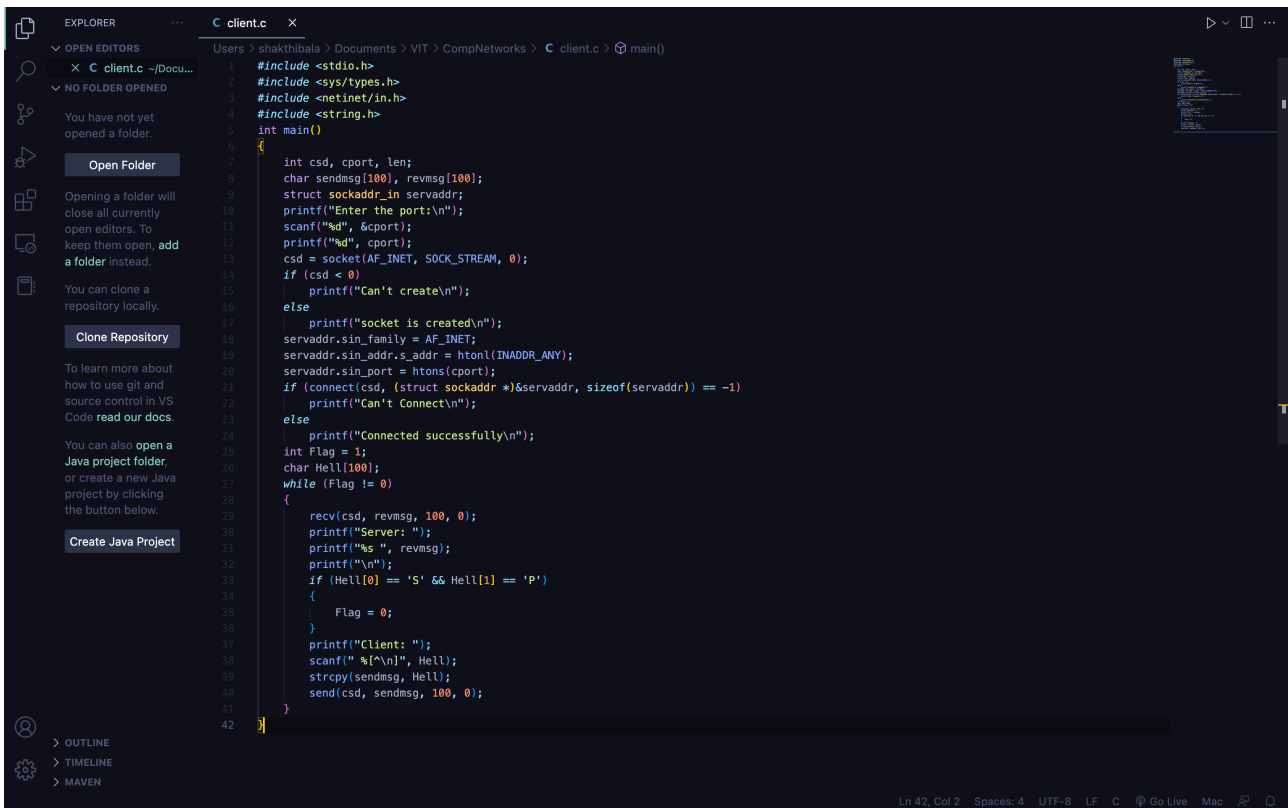
## LAB ASSESSMENT - 3

Program:

Server Side:

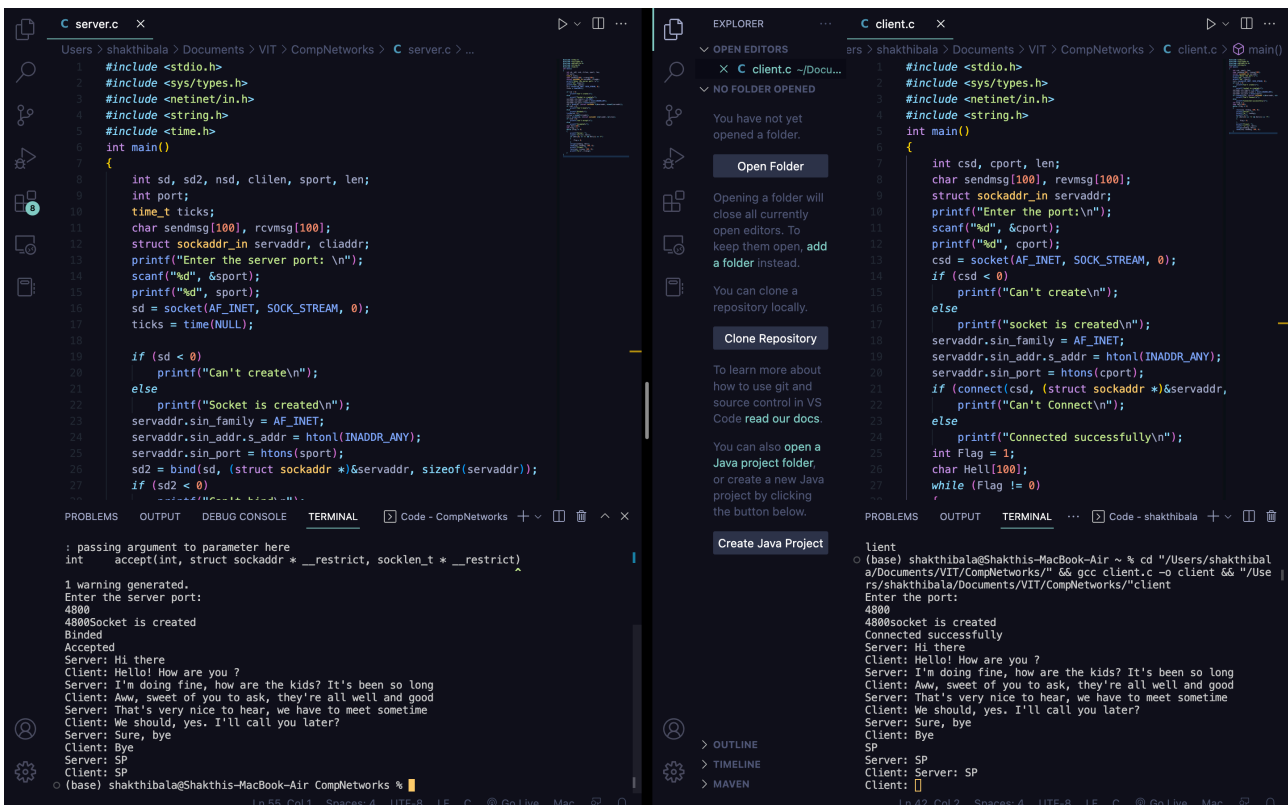
```
C server.c x
Users > shakthibala > Documents > VIT > CompNetworks > C server.c > ...
1  #include <stdio.h>
2  #include <sys/types.h>
3  #include <netinet/in.h>
4  #include <string.h>
5  #include <time.h>
6  int main()
7  {
8      int sd, sd2, nsd, clien, sport, len;
9      int port;
10     time_t ticks;
11     char sendmsg[100], rcvmsg[100];
12     struct sockaddr_in servaddr, cliaddr;
13     printf("Enter the server port: \n");
14     scanf("%d", &sport);
15     printf("%d", &sport);
16     sd = socket(AF_INET, SOCK_STREAM, 0);
17     ticks = time(NULL);
18
19     if (sd < 0)
20         printf("Can't create\n");
21     else
22         printf("Socket is created\n");
23     servaddr.sin_family = AF_INET;
24     servaddr.sin_addr.s_addr = htonl(INADDR_ANY);
25     servaddr.sin_port = htons(sport);
26     sd2 = bind(sd, (struct sockaddr *)&servaddr, sizeof(servaddr));
27     if (sd2 < 0)
28         printf("Can't bind\n");
29     else
30         printf("Binded\n");
31     listen(sd, 5);
32     clien = sizeof(cliaddr);
33     nsd = accept(sd, (struct sockaddr *)&cliaddr, &clien);
34     if (nsd < 0)
35         printf("can't accept\n");
36     else
37         printf("Accepted\n");
38     char Hell[100];
39     int Flag = 1;
40     while (Flag != 0)
41     {
42         printf("Server: ");
43         scanf("%[^\n]", Hell);
44         if (Hell[0] == 'S' && Hell[1] == 'P')
45         {
46             Flag = 0;
47
48             strcpy(sendmsg, Hell);
49             send(nsd, sendmsg, 100, 0);
50             printf("Client: ");
51             recv(nsd, rcvmsg, 100, 0);
52             printf("%s\n", rcvmsg);
53         }
54     }
55 }
```

# Client Side:



```
1 #include <stdio.h>
2 #include <sys/types.h>
3 #include <netinet/in.h>
4 #include <string.h>
5 int main()
6 {
7     int csd, cport, len;
8     char sendmsg[100], revmsg[100];
9     struct sockaddr_in servaddr;
10    printf("Enter the port:\n");
11    scanf("%d", &cport);
12    printf("%d", cport);
13    csd = socket(AF_INET, SOCK_STREAM, 0);
14    if (csd < 0)
15        printf("Can't create\n");
16    else
17        printf("socket is created\n");
18    servaddr.sin_family = AF_INET;
19    servaddr.sin_addr.s_addr = htonl(INADDR_ANY);
20    servaddr.sin_port = htons(cport);
21    if (connect(csd, (struct sockaddr *)&servaddr, sizeof(servaddr)) == -1)
22        printf("Can't Connect\n");
23    else
24        printf("Connected successfully\n");
25    int Flag = 1;
26    char Hell[100];
27    while (Flag != 0)
28    {
29        recv(csd, revmsg, 100, 0);
30        printf("Server: ");
31        printf("%s ", revmsg);
32        printf("\n");
33        if (Hell[0] == 'S' && Hell[1] == 'P')
34        {
35            Flag = 0;
36        }
37        printf("Client: ");
38        scanf("%s", Hell);
39        strcpy(sendmsg, Hell);
40        send(csd, sendmsg, 100, 0);
41    }
42 }
```

# Chat Application Execution:



```
1 #include <stdio.h>
2 #include <sys/types.h>
3 #include <netinet/in.h>
4 #include <string.h>
5 #include <time.h>
6 int main()
7 {
8     int sd, sd2, nsd, clien, sport, len;
9     int port;
10    time_t ticks;
11    char sendmsg[100], rcvmsg[100];
12    struct sockaddr_in servaddr, cliaddr;
13    printf("Enter the server port: \n");
14    scanf("%d", &sport);
15    printf("%d", sport);
16    sd = socket(AF_INET, SOCK_STREAM, 0);
17    ticks = time(NULL);
18
19    if (sd < 0)
20        printf("Can't create\n");
21    else
22        printf("Socket is created\n");
23    servaddr.sin_family = AF_INET;
24    servaddr.sin_addr.s_addr = htonl(INADDR_ANY);
25    servaddr.sin_port = htons(sport);
26    sd2 = bind(sd, (struct sockaddr *)&servaddr, sizeof(servaddr));
27    if (sd2 < 0)
28        printf("Can't bind\n");
29    else
30        printf("Server is listening\n");
31    while (1)
32    {
33        clien = accept(sd, NULL, NULL);
34        if (clien < 0)
35            continue;
36        printf("Client: %s", revmsg);
37        printf("\n");
38        if (Hell[0] == 'S' && Hell[1] == 'P')
39        {
40            Flag = 0;
41        }
42        printf("Client: ");
43        scanf("%s", Hell);
44        strcpy(sendmsg, Hell);
45        send(csd, sendmsg, 100, 0);
46    }
47 }
```