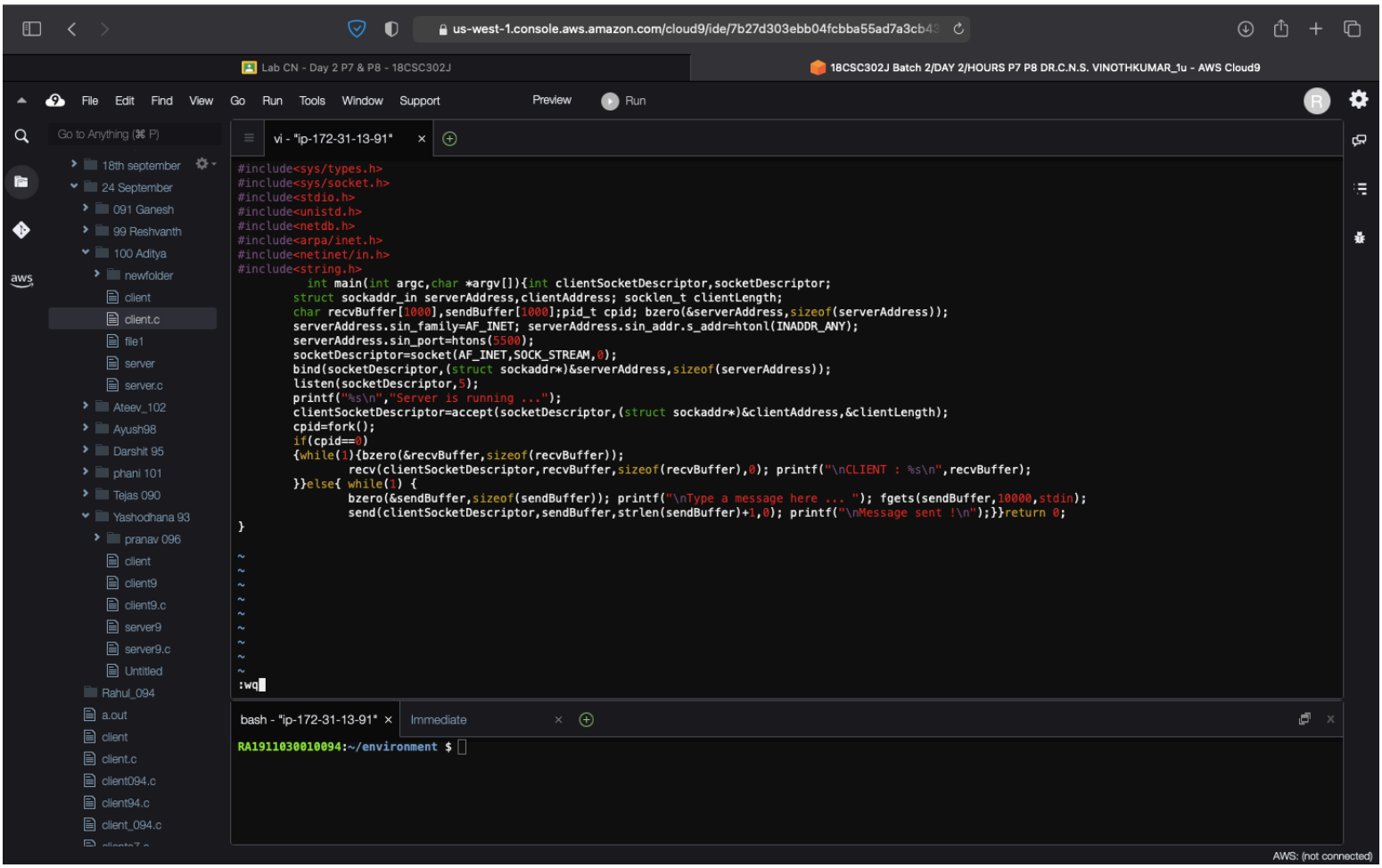
Computer Networks Lab - Experiment 7

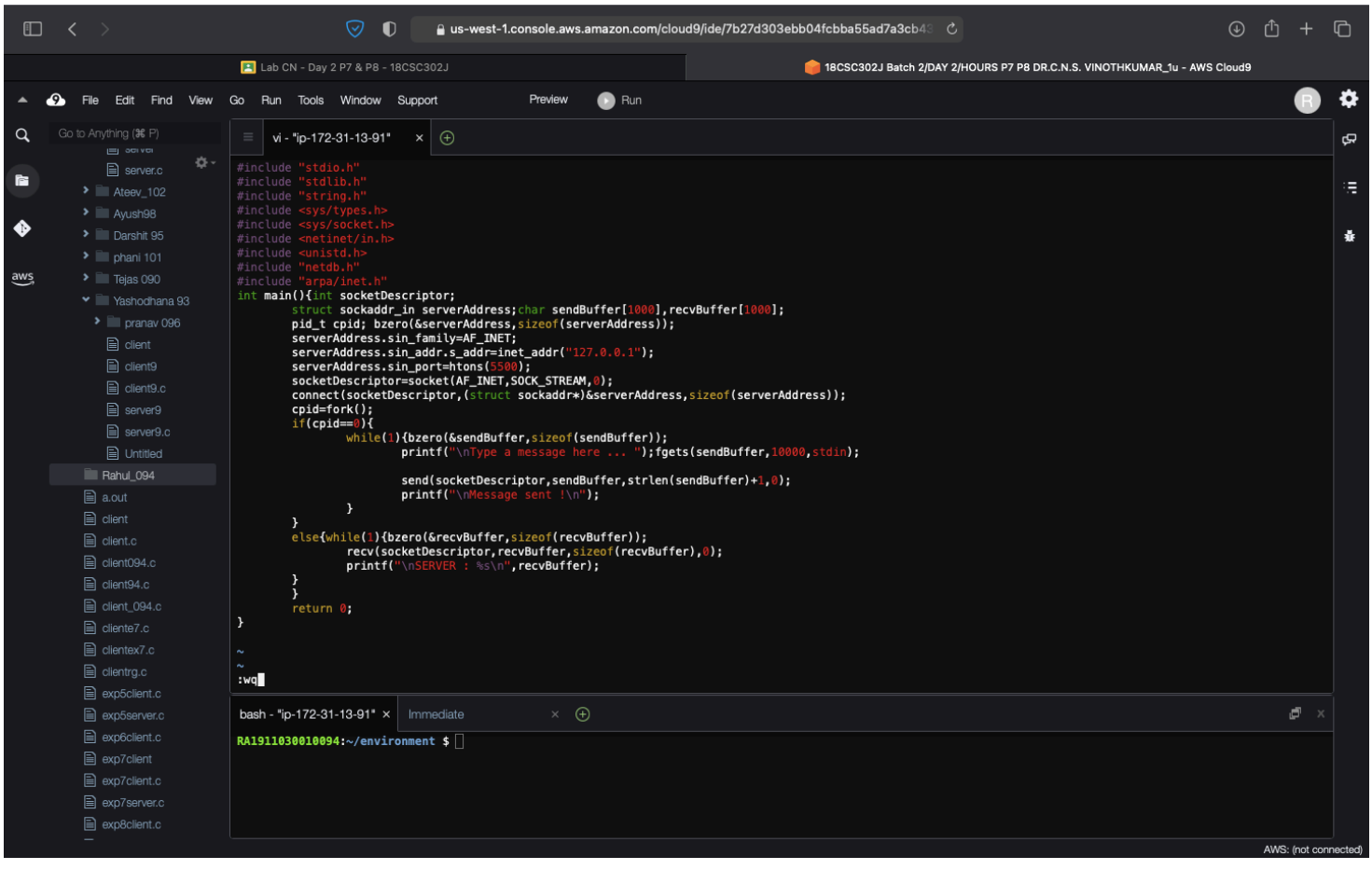
Name : Rahul Goel
Reg No: RA1911030010094
Section : O2

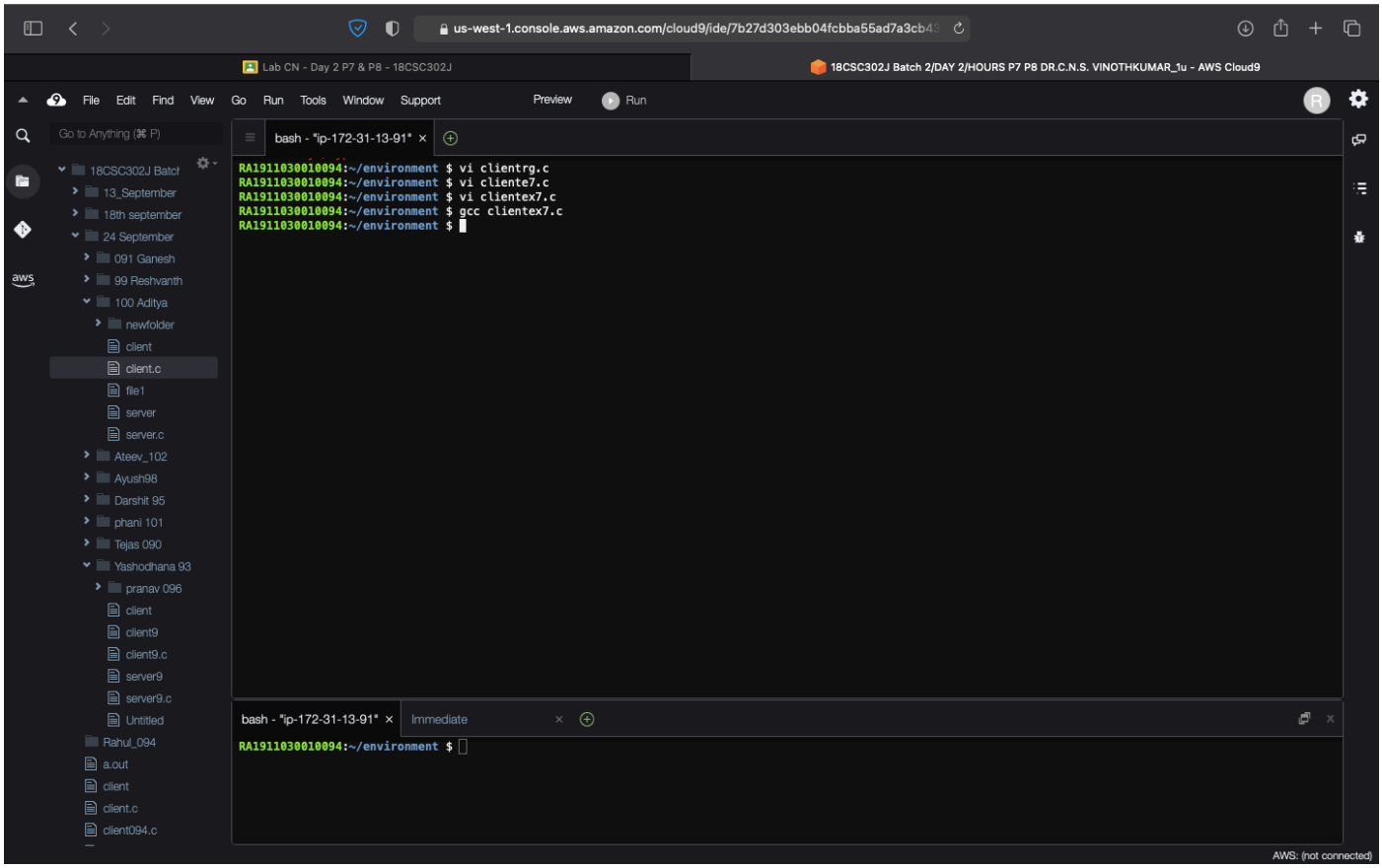
CODE: Server.c

```
#include<sys/types.h> #include<sys/socket.h> #include<stdio.h>
#include<unistd.h> #include<netdb.h> #include<arpa/inet.h>
#include<netinet/in.h> #include<string.h>

int main(int argc,char *argv[])
{
int clientSocketDescriptor,socketDescriptor;

struct sockaddr_in serverAddress,clientAddress; socklen_t clientLength;

char recvBuffer[1000],sendBuffer[1000];
pid_t cpid; bzero(&serverAddress,sizeof(serverAddress)); /*Socket
address structure*/ serverAddress.sin_family=AF_INET;
serverAddress.sin_addr.s_addr=htonl(INADDR_ANY);
serverAddress.sin_port=htons(5500);

/*TCP socket is created, an Internet socket address structure is filled with
wildcard address & server's well known port*/
socketDescriptor=socket(AF_INET,SOCK_STREAM,0);
/*Bind function assigns a local protocol address to the socket*/
bind(socketDescriptor,(struct
sockaddr*)&serverAddress,sizeof(serverAddress)); /*Listen function
specifies the maximum number of connections that kernel should queue

for this socket*/
listen(socketDescriptor,5);
printf("%s\n","Server is running ...");
/*The server to return the next completed connection from the front of the
completed connection Queue calls it*/
clientSocketDescriptor=accept(socketDescriptor,(struct
sockaddr*)&clientAddress,&clientLength);
```

```
/*Fork system call is used to create a new process*/
cpid=fork();

if(cpid==0)
{
while(1)
{
bzero(&recvBuffer,sizeof(recvBuffer));
/*Receiving the request from client*/
recv(clientSocketDescriptor,recvBuffer,sizeof(recvBuffer),0);
printf("\nCLIENT : %s\n",recvBuffer);

}
}
else
{ while(1) {

bzero(&sendBuffer,sizeof(sendBuffer)); printf("\nType a message here ...
"); /*Read the message from client*/ fgets(sendBuffer,10000,stdin);

/*Sends the message to client*/
send(clientSocketDescriptor,sendBuffer,strlen(sendBuffer)+1,0);
printf("\nMessage sent !\n");
}
}
return 0;

}
```


us-west-1.console.aws.amazon.com/cloud9/ide/7b27d303ebb04fcbba55ad7a3cb4
Lab CN - Day 2 P7 & P8 - 18CSC302J
18CSC302J Batch 2/DAY 2/HOURS P7 P8 DR.C.N.S. VINOTHKUMAR_1u - AWS Cloud9
File Edit Find View Go Run Tools Window Support Preview Run
Go to Anything (⌘ P)
18th september
24 September
091 Ganesh
99 Reshvanth
100 Aditya
newfolder
client
client.c
file1
server
server.c
Ateev_102
Ayush98
Darshit 95
phani 101
Tejas 090
Yashodhana 93
pranav 096
client
client9
client9.c
server9
server9.c
Untitled
Rahul_094
a.out
client
client.c
client094.c
client94.c
client_094.c
vi - "ip-172-31-13-91"
#include<sys/types.h>
#include<sys/socket.h>
#include<stdio.h>
#include<unistd.h>
#include<netdb.h>
#include<arpa/inet.h>
#include<netinet/in.h>
#include<string.h>
int main(int argc,char *argv[]){int clientSocketDescriptor,socketDescriptor;
struct sockaddr_in serverAddress,clientAddress; socklen_t clientLength;
char recvBuffer[1000],sendBuffer[1000];pid_t cpid; bzero(&serverAddress,sizeof(serverAddress));
serverAddress.sin_family=AF_INET; serverAddress.sin_addr.s_addr=htonl(INADDR_ANY);
serverAddress.sin_port=htons(5500);
socketDescriptor=socket(AF_INET,SOCK_STREAM,0);
bind(socketDescriptor,(struct sockaddr*)&serverAddress,sizeof(serverAddress));
listen(socketDescriptor,5);
printf("%s\n","Server is running ...");
clientSocketDescriptor=accept(socketDescriptor,(struct sockaddr*)&clientAddress,&clientLength);
cpid=fork();
if(cpid==0)
{while(1){bzero(&recvBuffer,sizeof(recvBuffer));
recv(clientSocketDescriptor,recvBuffer,sizeof(recvBuffer),0); printf("\nCLIENT : %s\n",recvBuffer);
}}else{ while(1) {
bzero(&sendBuffer,sizeof(sendBuffer)); printf("\nType a message here ... "); fgets(sendBuffer,10000,stdin);
send(clientSocketDescriptor,sendBuffer,strlen(sendBuffer)+1,0); printf("\nMessage sent !\n");}}return 0;
}
:wq
bash - "ip-172-31-13-91"
Immediate
RA1911030010094:~/environment $
AWS: (not connected)


us-west-1.console.aws.amazon.com/cloud9/ide/7b27d303ebb04fcbba55ad7a3cb4
Lab CN - Day 2 P7 & P8 - 18CSC302J
18CSC302J Batch 2/DAY 2/HOURS P7 P8 DR.C.N.S. VINOTHKUMAR_1u - AWS Cloud9
File Edit Find View Go Run Tools Window Support Preview Run
Go to Anything (⌘ P)
server.c
Ateev_102
Ayush98
Darshit 95
phani 101
Tejas 090
Yashodhana 93
pranav 096
client
client9
client9.c
server9
server9.c
Untitled
Rahul_094
a.out
client
client.c
client094.c
client94.c
client_094.c
cliente7.c
clientex7.c
clientrg.c
exp5client.c
exp5server.c
exp6client.c
exp7client
exp7client.c
exp7server.c
exp8client.c
vi - "ip-172-31-13-91"
#include "stdio.h"
#include "stdlib.h"
#include "string.h"
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <unistd.h>
#include "netdb.h"
#include "arpa/inet.h"
int main(){int socketDescriptor;
struct sockaddr_in serverAddress;char sendBuffer[1000],recvBuffer[1000];
pid_t cpid; bzero(&serverAddress,sizeof(serverAddress));
serverAddress.sin_family=AF_INET;
serverAddress.sin_addr.s_addr=inet_addr("127.0.0.1");
serverAddress.sin_port=htons(5500);
socketDescriptor=socket(AF_INET,SOCK_STREAM,0);
connect(socketDescriptor,(struct sockaddr*)&serverAddress,sizeof(serverAddress));
cpid=fork();
if(cpid==0){
while(1){bzero(&sendBuffer,sizeof(sendBuffer));
printf("\nType a message here ... ");fgets(sendBuffer,10000,stdin);
send(socketDescriptor,sendBuffer,strlen(sendBuffer)+1,0);
printf("\nMessage sent !\n");
}
}
else{while(1){bzero(&recvBuffer,sizeof(recvBuffer));
recv(socketDescriptor,recvBuffer,sizeof(recvBuffer),0);
printf("\nSERVER : %s\n",recvBuffer);
}
}
return 0;
}
:wq
bash - "ip-172-31-13-91"
Immediate
RA1911030010094:~/environment $
AWS: (not connected)


us-west-1.console.aws.amazon.com/cloud9/ide/7b27d303ebb04fcbba55ad7a3cb43
Lab CN - Day 2 P7 & P8 - 18CSC302J
18CSC302J Batch 2/DAY 2/HOURS P7 P8 DR.C.N.S. VINOTHKUMAR_1u - AWS Cloud9
File Edit Find View Go Run Tools Window Support
Preview
Run
Go to Anything (⌘ P)
18CSC302J Batch
13_September
18th september
24 September
091 Ganesh
99 Reshvanth
100 Aditya
newfolder
client
client.c
file1
server
server.c
Ateev_102
Ayush98
Darshit 95
phani 101
Tejas 090
Yashodhana 93
pranav 096
client
client9
client9.c
server9
server9.c
Untitled
Rahul_094
a.out
client
client.c
client094.c
bash - "ip-172-31-13-91"
RA1911030010094:~/environment $ vi clientrg.c
RA1911030010094:~/environment $ vi cliente7.c
RA1911030010094:~/environment $ vi clientex7.c
RA1911030010094:~/environment $ gcc clientex7.c
RA1911030010094:~/environment $
bash - "ip-172-31-13-91"
Immediate
RA1911030010094:~/environment $
AWS: (not connected)

RESULT:

Thus, a full duplex chat program where server sends a message and client prints it and vice versa using TCP/IP is written and executed successfully.