Experiement 4:

Server Code :

```
#include<sys/socket.h>
#include<stdio.h>
#include<stdlib.h>
#include<unistd.h>
#include<string.h>
#include<netinet/in.h>
#include<netdb.h>
#include<arpa/inet.h>
#include<sys/types.h>
int main(int argc,char *argv[])
{
int sd;
char buff[1024];
struct sockaddr_in cliaddr,servaddr;
socklen_t clilen;
clilen=sizeof(cliaddr);
sd=socket(AF_INET,SOCK_DGRAM,0);
if (sd<0)
{
perror ("Cannot open Socket");
exit(1);
}
bzero(&servaddr,sizeof(servaddr));
servaddr.sin_family=AF_INET;
servaddr.sin_addr.s_addr=htonl(INADDR_ANY);
servaddr.sin_port=htons(9000);

if(bind(sd,(struct sockaddr*)&servaddr,sizeof(servaddr))<0)
{
perror("error in binding the port");
exit(1);
}
printf("%s","Server is Running…\n");
while(1)
{
bzero(&buff,sizeof(buff));
if(recvfrom(sd,buff,sizeof(buff),0,(struct sockaddr*)&cliaddr,&clilen)<0)
{
perror("Cannot rec data");
exit(1);
 }
printf("Message is received \n",buff);
if(sendto(sd,buff,sizeof(buff),0,(struct sockadddr*)&cliaddr,clilen)<0)
  {
perror("Cannot send data to client");
exit(1);
          }
   printf("Send data to UDP Client: %s",buff);
}
close(sd);
return 0;
}
```

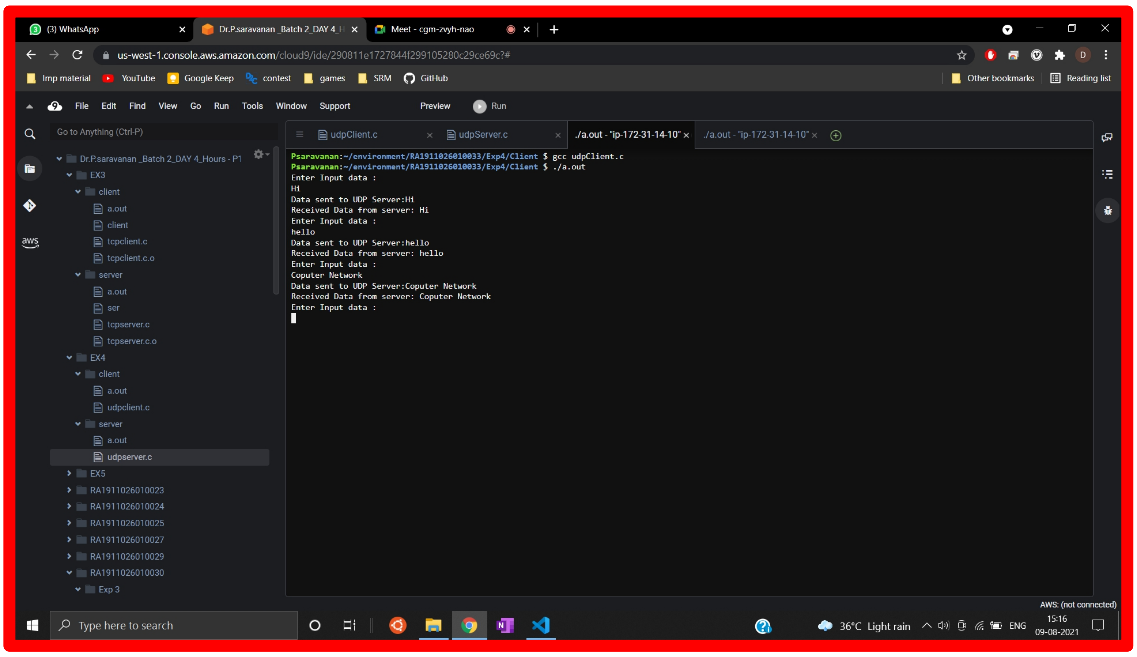
Client Code:

```
#include<sys/types.h>
#include<sys/socket.h>
#include<stdio.h>
#include<unistd.h>
#include<string.h>
#include<netinet/in.h>
#include<stdlib.h>
#include<netdb.h>
int main(int argc,char*argv[])
{
int sd;
char buff[1024];
struct sockaddr_in servaddr;
socklen_t len;
len=sizeof(servaddr);

sd = socket(AF_INET,SOCK_DGRAM,0);
if(sd<0)
{
perror("Cannot open socket");
exit(1);
}
bzero(&servaddr,len);
servaddr.sin_family=AF_INET;
servaddr.sin_addr.s_addr=htonl(INADDR_ANY);
servaddr.sin_port=htons(9000);
while(1)
{
printf("Enter Input data : \n");
bzero(buff,sizeof(buff));
fgets(buff,sizeof (buff),stdin);
if(sendto (sd,buff,sizeof (buff),0,(struct sockaddr*)&servaddr,len)<0)
    {
          perror("Cannot send data");
          exit(1);
     }
printf("Data sent to UDP Server:%s",buff);
bzero(buff,sizeof(buff));
if(recvfrom (sd,buff,sizeof(buff),0,(struct sockaddr*)&servaddr,&len)<0)
     {
          perror("Cannot receive data");
         exit(1);
     }
printf("Received Data from server: %s",buff);
  }
close(sd);
return 0;
}
```

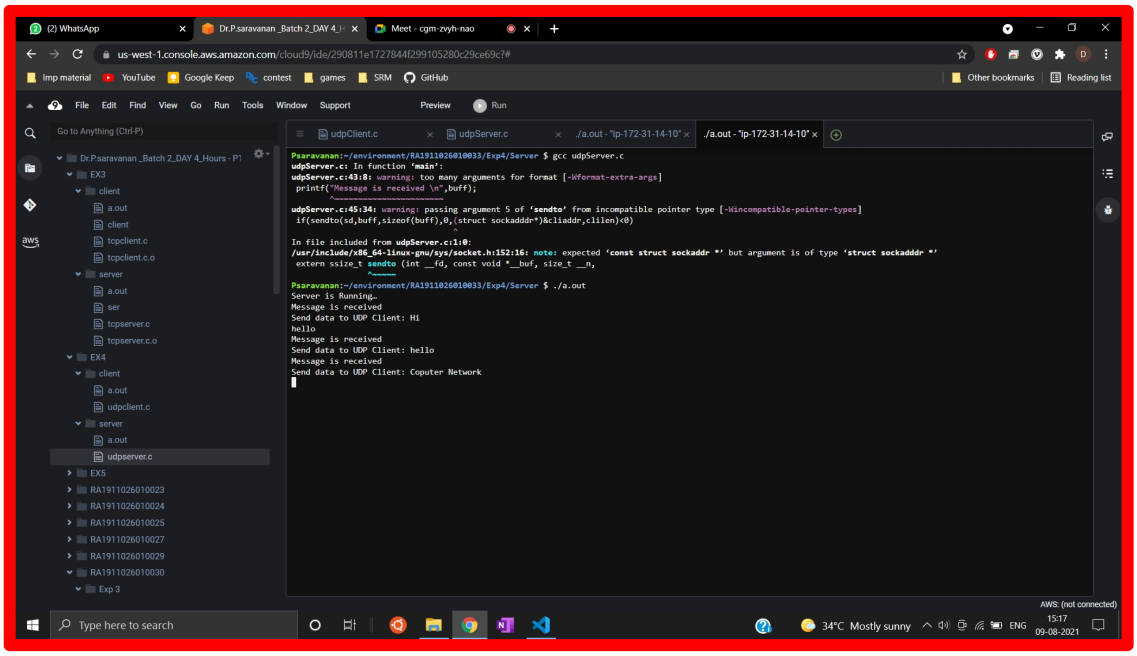
Output :

Client :



```
Psaravanan:~/environment/RA1911026010033/Exp4/Client $ gcc udpClient.c
Psaravanan:~/environment/RA1911026010033/Exp4/Client $ ./a.out
Enter Input data :
Hi
Data sent to UDP Server:Hi
Received Data from server: Hi
Enter Input data :
hello
Data sent to UDP Server:hello
Received Data from server: hello
Enter Input data :
Coputer Network
Data sent to UDP Server:Coputer Network
Received Data from server: Coputer Network
Enter Input data :
```

Server:



```
Psaravanan:~/environment/RA1911026010033/Exp4/Server $ gcc udpServer.c
udpServer.c: In function ‘main’:
udpServer.c:43:8: warning: too many arguments for format [-Wformat-extra-args]
 printf("Message is received \n",buff);
        ^~~~~~~~~~~~~~~~~~~~~~~~
udpServer.c:45:34: warning: passing argument 5 of ‘sendto’ from incompatible pointer type [-Wincompatible-pointer-types]
 if(sendto(sd,buff,sizeof(buff),0,(struct sockadddr*)&cliaddr,clilen)<0)
                                  ^
In file included from udpServer.c:1:0:
/usr/include/x86_64-linux-gnu/sys/socket.h:152:16: note: expected ‘const struct sockaddr *’ but argument is of type ‘struct sockadddr *’
 extern ssize_t sendto (int __fd, const void *__buf, size_t __n,
                ^~~~~~
Psaravanan:~/environment/RA1911026010033/Exp4/Server $ ./a.out
Server is Running…
Message is received
Send data to UDP Client: Hi
hello
Message is received
Send data to UDP Client: hello
Message is received
Send data to UDP Client: Coputer Network
```