Lab Ex.4 - (Client / Server communication using UDP) Experiment

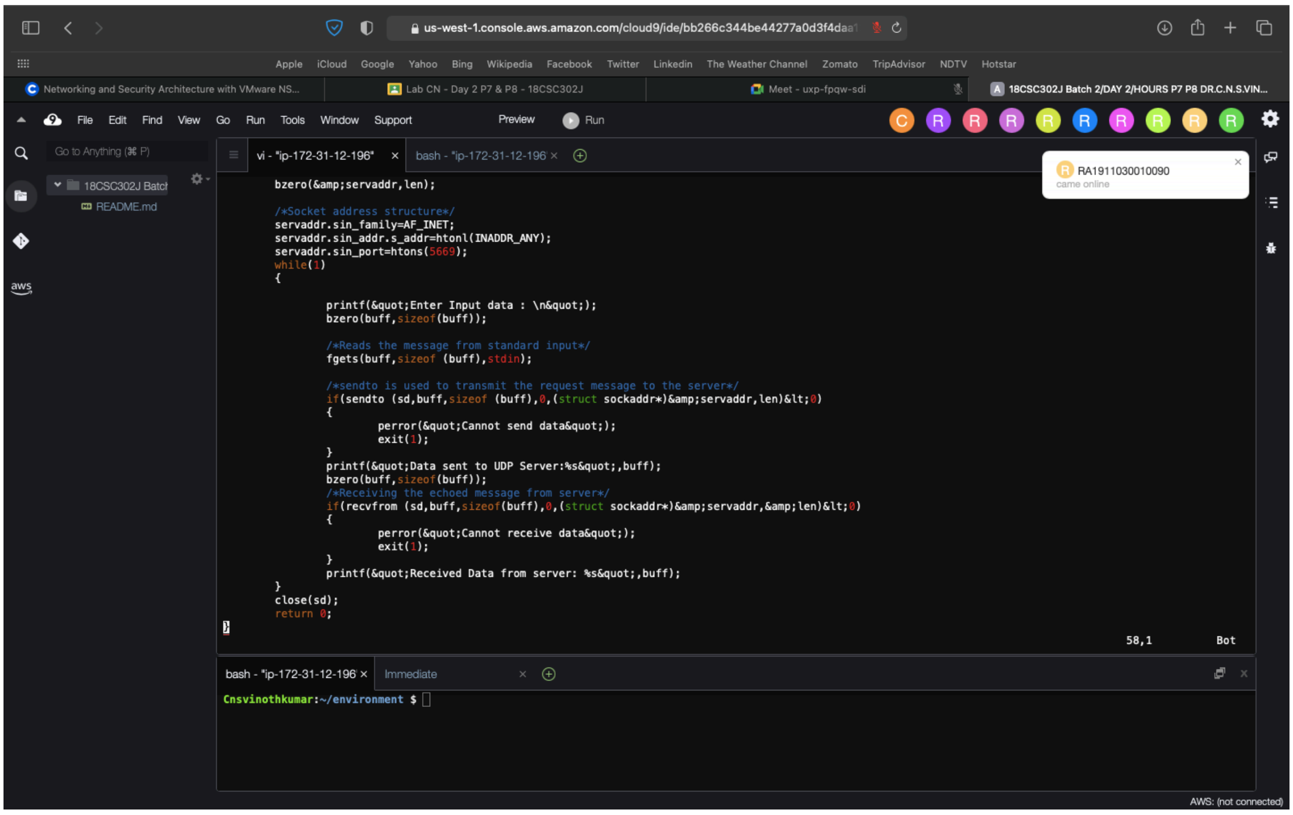
Name: Rahul Goel

Reg.no: RA1911030010094

SERVER CODE:

```
#include <stdio.h> #include <stdlib.h> #include <unistd.h> #include <string.h>
#include <sys/types.h> #include <sys/socket.h> #include <arpa/inet.h> #include
<netinet/in.h> #define PORT 8080 #define MAXLINE 1024 int main() {

int sockfd;
char buffer[MAXLINE];
char *hello = "Hello from server";
struct sockaddr_in servaddr, cliaddr;
if ( (sockfd = socket(AF_INET, SOCK_DGRAM, 0)) < 0 ) { perror("socket
creation failed");
exit(EXIT_FAILURE);

}
memset(&servaddr, 0, sizeof(servaddr)); memset(&cliaddr, 0, sizeof(cliaddr));
servaddr.sin_family = AF_INET; servaddr.sin_addr.s_addr = INADDR_ANY;
servaddr.sin_port = htons(PORT);
if ( bind(sockfd, (const struct sockaddr *)&servaddr, sizeof(servaddr)) < 0 )
{
perror("bind failed");
exit(EXIT_FAILURE);
}
int len, n;
len = sizeof(cliaddr);
n = recvfrom(sockfd, (char *)buffer, MAXLINE, MSG_WAITALL, ( struct
sockaddr *) &cliaddr, &len);
buffer[n] = '\0';
printf("Client : %s\n", buffer);
sendto(sockfd, (const char *)hello, strlen(hello), MSG_CONFIRM, (const struct
sockaddr *) &cliaddr, len);
```

printf("Hello message sent.\n");
return 0;
}



```
	bzero(&amp;servaddr,len);

	/*Socket address structure*/
	servaddr.sin_family=AF_INET;
	servaddr.sin_addr.s_addr=htonl(INADDR_ANY);
	servaddr.sin_port=htons(5669);
	while(1)
	{

		printf(&quot;Enter Input data : \n&quot;);
		bzero(buff,sizeof(buff));

		/*Reads the message from standard input*/
		fgets(buff,sizeof (buff),stdin);

		/*sendto is used to transmit the request message to the server*/
		if(sendto (sd,buff,sizeof (buff),0,(struct sockaddr*)&amp;servaddr,len)&lt;0)
		{
			perror(&quot;Cannot send data&quot;);
			exit(1);
		}
		printf(&quot;Data sent to UDP Server:%s&quot;,buff);
		bzero(buff,sizeof(buff));
		/*Receiving the echoed message from server*/
		if(recvfrom (sd,buff,sizeof(buff),0,(struct sockaddr*)&amp;servaddr,&amp;len)&lt;0)
		{
			perror(&quot;Cannot receive data&quot;);
			exit(1);
		}
		printf(&quot;Received Data from server: %s&quot;,buff);
	}
	close(sd);
	return 0;
}
                                                                    58,1          Bot
```

```
Cnsvinothkumar:~/environment $
```

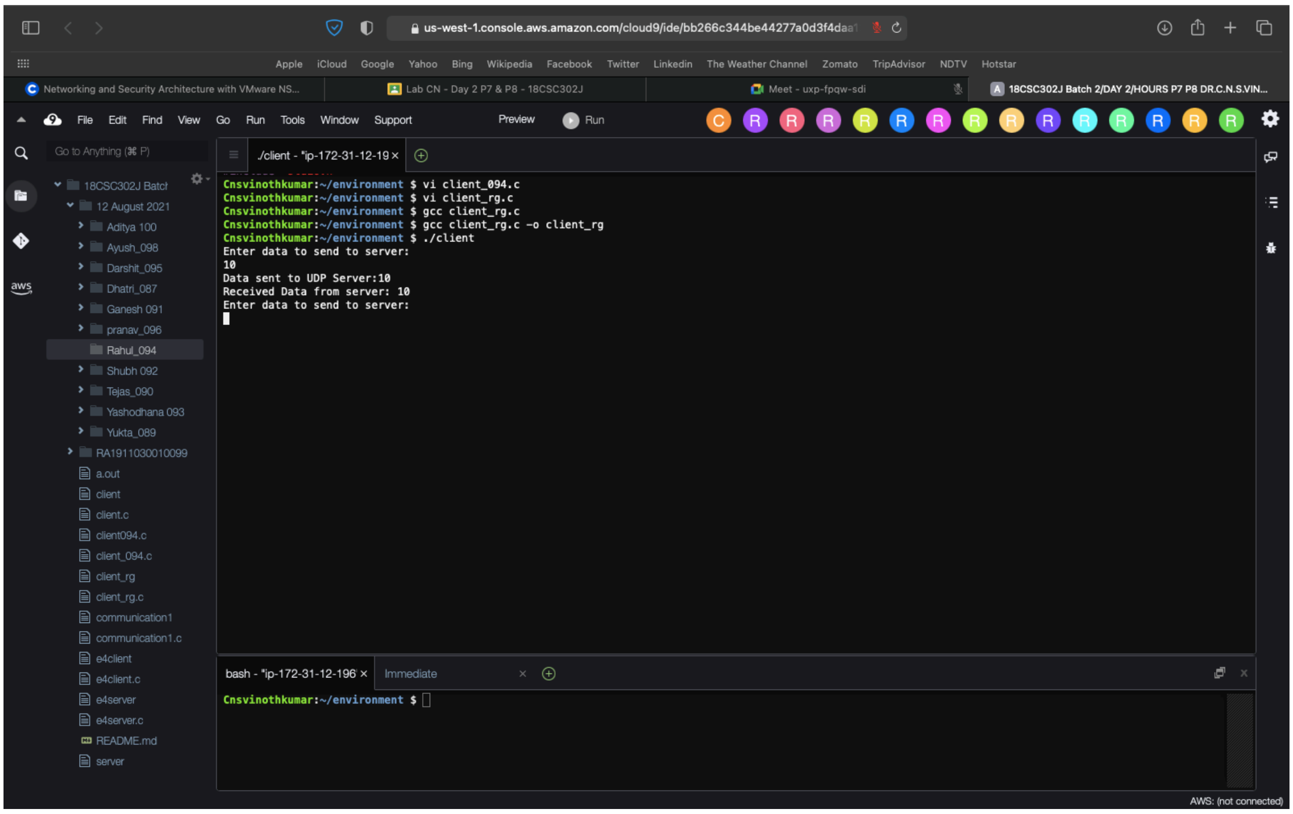
CLIENT CODE:

#include <stdio.h> #include <stdlib.h> #include <unistd.h> #include <string.h> #include <sys/types.h> #include <sys/socket.h> #include <arpa/inet.h> #include <netinet/in.h> #define PORT 8080 #define MAXLINE 1024 int main() {

int sockfd;
char buffer[MAXLINE];
char *hello = "Hello from client"; struct sockaddr_in servaddr;

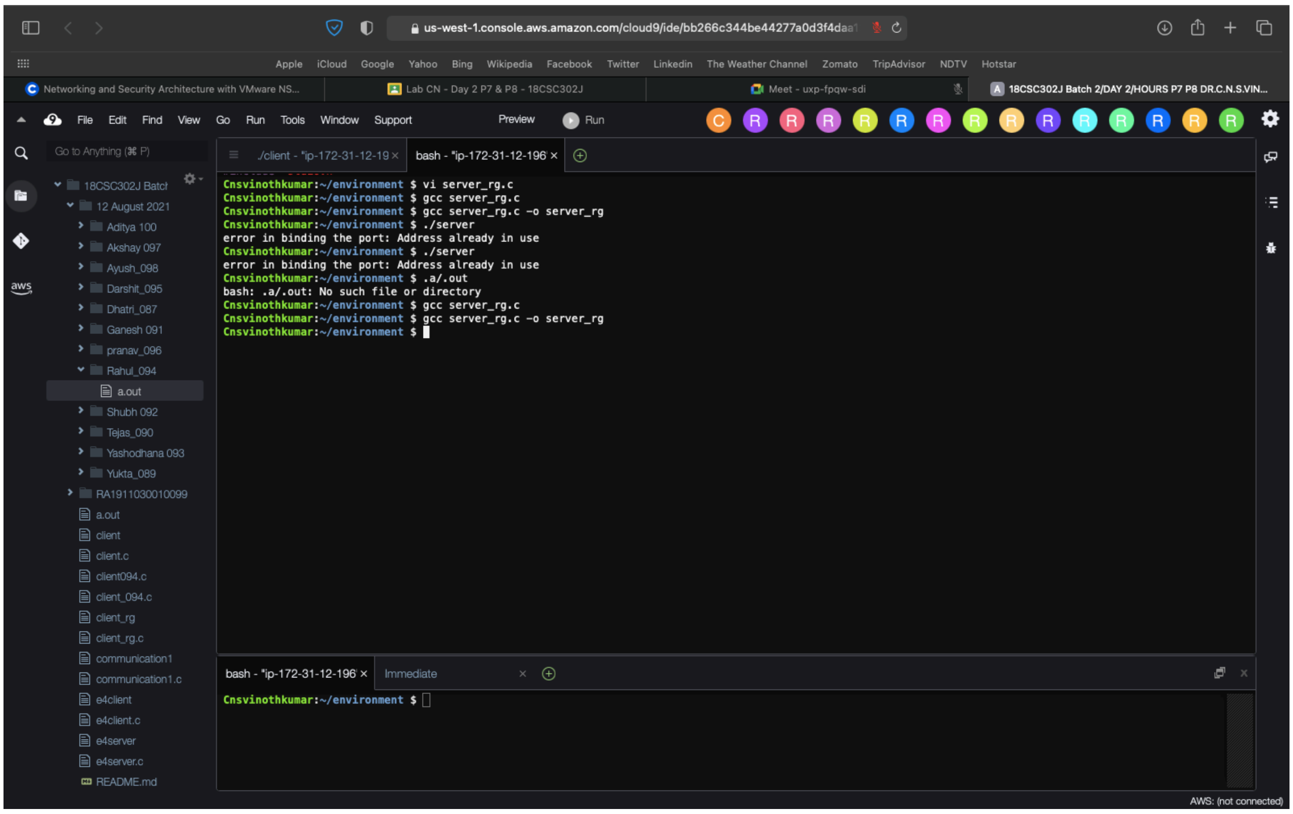
if ((sockfd = socket(AF_INET, SOCK_DGRAM, 0)) < 0) { perror("socket creation failed");
exit(EXIT_FAILURE);
}

```
memset(&servaddr, 0, sizeof(servaddr)); servaddr.sin_family = AF_INET;
servaddr.sin_port = htons(PORT); servaddr.sin_addr.s_addr = INADDR_ANY;
int n, len;

sendto(sockfd, (const char *)hello, strlen(hello), MSG_CONFIRM, (const struct
sockaddr *) &servaddr, sizeof(servaddr));
printf("Hello message sent.\n");

n = recvfrom(sockfd, (char *)buffer, MAXLINE, MSG_WAITALL, (struct
sockaddr *) &servaddr, &len);
buffer[n] = '\0';

printf("Server : %s\n", buffer); close(sockfd);
return 0;
}
```



```
Cnsvinothkumar:~/environment $ vi client_094.c
Cnsvinothkumar:~/environment $ vi client_rg.c
Cnsvinothkumar:~/environment $ gcc client_rg.c
Cnsvinothkumar:~/environment $ gcc client_rg.c -o client_rg
Cnsvinothkumar:~/environment $ ./client
Enter data to send to server:
10
Data sent to UDP Server:10
Received Data from server: 10
Enter data to send to server:
```

```
Cnsvinothkumar:~/environment $
```



```
Cnsvinothkumar:~/environment $ vi server_rg.c
Cnsvinothkumar:~/environment $ gcc server_rg.c
Cnsvinothkumar:~/environment $ gcc server_rg.c -o server_rg
Cnsvinothkumar:~/environment $ ./server
error in binding the port: Address already in use
Cnsvinothkumar:~/environment $ ./server
error in binding the port: Address already in use
Cnsvinothkumar:~/environment $ .a/.out
bash: .a/.out: No such file or directory
Cnsvinothkumar:~/environment $ gcc server_rg.c
Cnsvinothkumar:~/environment $ gcc server_rg.c -o server_rg
Cnsvinothkumar:~/environment $
```

```
Cnsvinothkumar:~/environment $
```