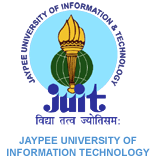
**SYSTEM AND NETWORKS PROGRAMMING**

**ONLINE DRESS & WATCH SHOPPING SYSTEM**

FACULTY INCHARGE: SUMAN SAHA



**SUBMITTED BY:**

**AMY MEHNDIRATTA 111240**

**SHUBHI JAIN 111338**

**Description:**

To implement a online dress and watch shopping system which provides the following functionalities (services) to the customer:

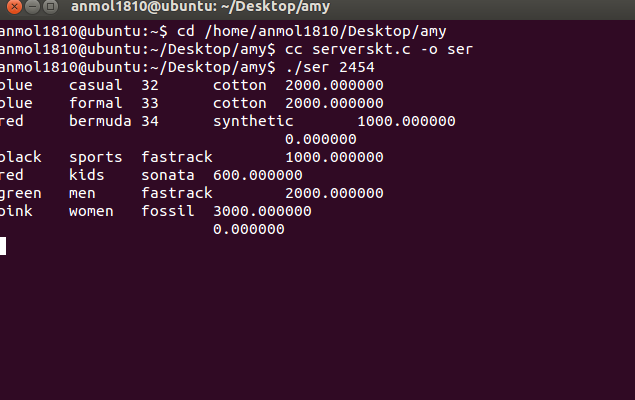
* View the list of an item and cost for a particular type chosen
* Order of a particular item or list of items ordered
* Search for ordered list
* Wish list of the customer

**Objectives:**

* To discuss the client and server communication
* To develop an application for client to client communication and client to server communication
* To perform basic operations of searching , ordering and tracing of an item(s) ordered online through client server mechanism

**CODE:**

**Serverskt.c**



/\* A simple server in the internet domain using TCP

The port number is passed as an argument \*/

#include "structure.h"

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <unistd.h>

#include <sys/types.h>

#include <sys/socket.h>

#include <netinet/in.h>

typedef struct

{

APP app[10];

APP app1[20];

}APPLICATION;

void error(const char \*msg)

{

perror(msg);

exit(1);

}

int main(int argc, char \*argv[])

{

APPLICATION ap;

APPLICATION ap1;

int client;

int login=0;

APP a[10];

APP c[10];

APP e[10];

APP b;

APP f;

FILE \*fp1,\*fp2,\*fp3,\*fp4;

int sockfd, newsockfd, portno;

socklen\_t clilen;

char buffer[256];

struct sockaddr\_in serv\_addr, cli\_addr;

int n,i,flag=0,j=0,k=0;

fp1=fopen("user\_account","r");

i=0;

while(!feof(fp1))

{

fscanf(fp1,"%s%s",a[i].l.uname,a[i].l.pwd);

i++;

}

a[0].len=i-1;

{

fscanf(fp2,"%s%s%s%s%f",c[j].d.color,c[j].d.type,c[j].d.size,c[j].d.material,&c[j].d.cost);

j++;

}

c[0].d.len=j-1;

fp3=fopen("watch","r");

k=0;

while(!feof(fp3))

{

fscanf(fp3,"%s%s%s%f",e[k].w.color,e[k].w.type,e[k].w.make,&e[k].w.cost);

k++;

}

e[0].w.len=k-1;

printf("%d",e[0].w.len);

if (argc < 2) {

fprintf(stderr,"ERROR, no port provided\n");

exit(1);

}

sockfd = socket(AF\_INET, SOCK\_STREAM, 0);

if (sockfd < 0)

error("ERROR opening socket");

bzero((char \*) &serv\_addr, sizeof(serv\_addr));

portno = atoi(argv[1]);

serv\_addr.sin\_family = AF\_INET;

serv\_addr.sin\_addr.s\_addr = INADDR\_ANY;

serv\_addr.sin\_port = htons(portno);

if (bind(sockfd, (struct sockaddr \*) &serv\_addr,

sizeof(serv\_addr)) < 0)

error("ERROR on binding");

while(1)

{

listen(sockfd,15);

clilen = sizeof(cli\_addr);

newsockfd = accept(sockfd,

(struct sockaddr \*) &cli\_addr,

&clilen);

if (newsockfd < 0)

error("ERROR on accept");

bzero(buffer,256);

n = read(newsockfd,&b,sizeof(APP));

if (n < 0) error("ERROR reading from socket");

if(b.code==-33)

{

for(i=0;i<a[0].len;i++)

{

if(strcmp(b.l.uname,a[i].l.uname)==0 && strcmp(b.l.pwd,a[i].l.pwd)==0)

{

printf("\nlogged in\n");

b.code=1;

login=1;

client=i;

n = write(newsockfd,&b,sizeof(APP));

break;

}

}

}

while(login)

{

n = read(newsockfd,&b,sizeof(APP));

if(b.code==0)

{

login=0;

printf("\nlogged out\n");

break;

}

if(b.code==1)

{

for(i=0;i<c[0].d.len;i++)

{

ap.app[i]=c[i];

}

n = write(newsockfd,&ap,sizeof(APPLICATION));

}

if(b.code==2)

{

for(i=0;i<e[0].w.len;i++)

{

ap1.app1[i]=e[i];

}

n = write(newsockfd,&ap1,sizeof(APPLICATION));

}

if(b.code==3)

{

printf("\nsearching\n");

for(i=0;i<c[0].d.len;i++)

{

if(strcmp(b.d.type,c[i].d.type)==0)

{

printf("\nsearching successs\n");

b.d.cost=c[i].d.cost;

n = write(newsockfd,&b,sizeof(APP));

break;

}

}

}

if(b.code==4)

{

printf("\nsearching\n");

for(i=0;i<e[0].w.len;i++)

{

if(strcmp(b.w.type,e[i].w.type)==0)

{

printf("\nsearching successs\n");

b.w.cost=e[i].w.cost;

n = write(newsockfd,&b,sizeof(APP));

break;

}

}

}

if(b.code==5)

{

printf("%d",b.code);

ap.app[0].d.len=0;

for(i=0;i<c[0].d.len;i++)

{

if(strcmp(b.d.type,c[i].d.type)==0)

{

ap.app[i]=c[i];

ap.app[0].d.len+=1;

n = write(newsockfd,&ap,sizeof(APPLICATION));

break;

}

}

}

if(b.code==6)

{

ap.app[0].w.len=0;

for(i=0;i<e[0].w.len;i++)

{

if(strcmp(b.w.type,e[i].w.type)==0)

{

ap.app[i]=e[i];

ap.app[0].w.len+=1;

n = write(newsockfd,&ap,sizeof(APPLICATION));

printf("%d",n);

break;

}

}

}

}

close(newsockfd);

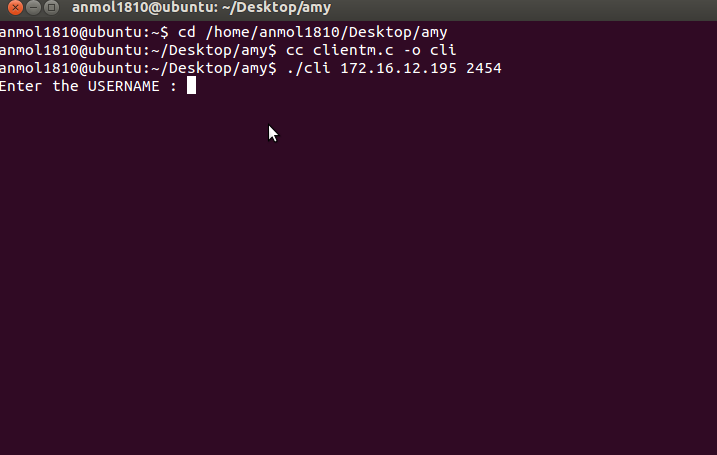
}

close(sockfd);

return 0;

}

**Clientm.c**



#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#include <string.h>

#include <sys/types.h>

#include <sys/socket.h>

#include <netinet/in.h>

#include <netdb.h>

#include "structure.h"

typedef struct

{

APP app[10];

APP app1[20];

}APPLICATION;

void error(const char \*msg)

{

perror(msg);

exit(0);

}

int main(int argc, char \*argv[])

{

APPLICATION ap;

APPLICATION ap1;

APP c;

APP c1;

APP c2[10];

int chance=3;

int sockfd, portno, n,i,n1;

struct sockaddr\_in serv\_addr;

struct hostent \*server;

char buffer[256];

if (argc < 3) {

fprintf(stderr,"usage %s hostname port\n", argv[0]);

exit(0);

}

portno = atoi(argv[2]);

sockfd = socket(AF\_INET, SOCK\_STREAM, 0);

if (sockfd < 0)

error("ERROR opening socket");

server = gethostbyname(argv[1]);

if (server == NULL) {

fprintf(stderr,"ERROR, no such host\n");

exit(0);

}

bzero((char \*) &serv\_addr, sizeof(serv\_addr));

serv\_addr.sin\_family = AF\_INET;

bcopy((char \*)server->h\_addr,

(char \*)&serv\_addr.sin\_addr.s\_addr,

server->h\_length);

serv\_addr.sin\_port = htons(portno);

if (connect(sockfd,(struct sockaddr \*) &serv\_addr,sizeof(serv\_addr)) < 0)

error("ERROR connecting");

while(chance>0)

{

printf("enter username : ");

scanf("%s",c.l.uname);

printf("enter PWD : ");

scanf("%s",c.l.pwd);

c.code=-33;

n = write(sockfd,&c,sizeof(APP));

n = read(sockfd,&c1,sizeof(APP));

if(c1.code==1)

{

printf("logged");

break;

}

chance--;

if(chance==0)

{

printf("\nexceeds login chances\n");

return 1;

}

}

while(1){

printf("\n1.List Dresses\n2.List Watches\n3.Cost for a particular type in Dresses\n4.Cost for a particular type in Watches\n5.Order for Dresses \n6.Order for watches\n\n7.Offers available\n8.LOGOUT\n");

printf("Please enter the selection:\n ");

scanf("%d",&c.code);

if(c.code==8)

{

printf("\nprogram exiting........\n");

c.code=0;

n=write(sockfd,&c,sizeof(APP));

close(sockfd);

break;

}

if(c.code==1)

{

n = write(sockfd,&c,sizeof(APP));

n = read(sockfd,&ap,sizeof(APPLICATION));

printf("\nCOLOR\tTYPE\tSIZE\tMATERIAL\tCOST\n");

for(i=0;i<ap.app[0].d.len;i++)

{

printf("%s\t%s\t%s\t%s\t%f\n",ap.app[i].d.color,ap.app[i].d.type,ap.app[i].d.size,ap.app[i].d.material,ap.app[i].d.cost);

}

}

if(c.code==2)

{

n = write(sockfd,&c,sizeof(APP));

n = read(sockfd,&ap1,sizeof(APPLICATION));

printf("\nCOLOR\tTYPE\tMAKE\tCOST\n");

printf("%d\n",ap1.app1[0].w.len);

for(i=0;i<ap1.app1[0].w.len;i++)

{

printf("%s\t%s\t%s\t%f\n",ap1.app1[i].w.color,ap1.app1[i].w.type,ap1.app1[i].w.make,ap1.app1[i].w.cost);

}

}

if(c.code==3)

{

printf("Enter the type of the dress you want to search for to find its cost\n");

scanf("%s",c.d.type);

n = write(sockfd,&c,sizeof(APP));

n = read(sockfd,&c,sizeof(APP));

printf("The cost of the type of the dress which you have searced for is %f\n",c.d.cost);

}

if(c.code==4)

{

printf("Enter the type of the watch you want to search for to find its cost\n");

scanf("%s",c.w.type);

n = write(sockfd,&c,sizeof(APP));

n = read(sockfd,&c,sizeof(APP));

printf("The cost of the type of the watch which you have searced for is %f\n",c.w.cost);

}

if(c.code==5)

{

printf("Enter the type of the dress you want to order\n");

scanf("%s",c.d.type);

n = write(sockfd,&c,sizeof(APP));

n = read(sockfd,&ap,sizeof(APP));

printf("The details of the ordered items are\n");

for(i=0;i<ap.app[0].d.len;i++)

{

printf("%s\t%s\t%s\t%s\t%f\n",ap.app[i].d.color,ap.app[i].d.type,ap.app[i].d.size,ap.app[i].d.material,ap.app[i].d.cost);

}

while(1)

{

printf("Whether you want to continue order the items\n");

printf("1.Yes and 2.NO\n");

scanf("%d",&n1);

if(n1==1)

{

printf("Enter the type of the dress you want to order\n");

scanf("%s",c.d.type);

n = write(sockfd,&c,sizeof(APP));

n = read(sockfd,&ap,sizeof(APP));

printf("The details of the ordered items are\n");

for(i=0;i<ap.app[0].d.len;i++)

{

printf("%s\t%s\t%s\t%s\t%f\n",ap.app[i].d.color,ap.app[i].d.type,ap.app[i].d.size,ap.app[i].d.material,ap.app[i].d.cost);

}

}

else

{

printf("The user doesn't want to order more items\n");

break;

}

}

}

if(c.code==6)

{

printf("Enter the type of the watch you want to order\n");

scanf("%s",c.w.type);

n = write(sockfd,&c,sizeof(APP));

n = read(sockfd,&ap,sizeof(APP));

printf("The details of the ordered items are\n");

printf("%d",ap.app[0].w.len);

for(i=0;i<ap.app[0].w.len;i++)

{

printf("%s\t%s\t%s\t%f\n",ap.app[i].w.color,ap.app[i].w.type,ap.app[i].w.make,ap.app[i].w.cost);

}

while(1)

{

printf("Whether you want to continue order the items\n");

printf("1.Yes and 2.NO\n");

scanf("%d",&n1);

if(n1==1)

{

printf("Enter the type of the watch you want to order\n");

scanf("%s",c.w.type);

n = write(sockfd,&c,sizeof(APP));

n = read(sockfd,&ap,sizeof(APP));

printf("The details of the ordered items are\n");

printf("%d",ap.app[0].w.len);

for(i=0;i<ap.app[0].w.len;i++)

{

printf("%s\t%s\t%s\t%f\n",ap.app[i].w.color,ap.app[i].w.type,ap.app[i].w.make,ap.app[i].w.cost);

}

}

else

{

printf("The user doesn't want to order more items\n");

break;

}

}

}

}

if (n < 0)

error("ERROR writing to socket");

bzero(buffer,256);

n = read(sockfd,&c,sizeof(APP));

if (n < 0)

error("ERROR reading from socket");

close(sockfd);

return 0;

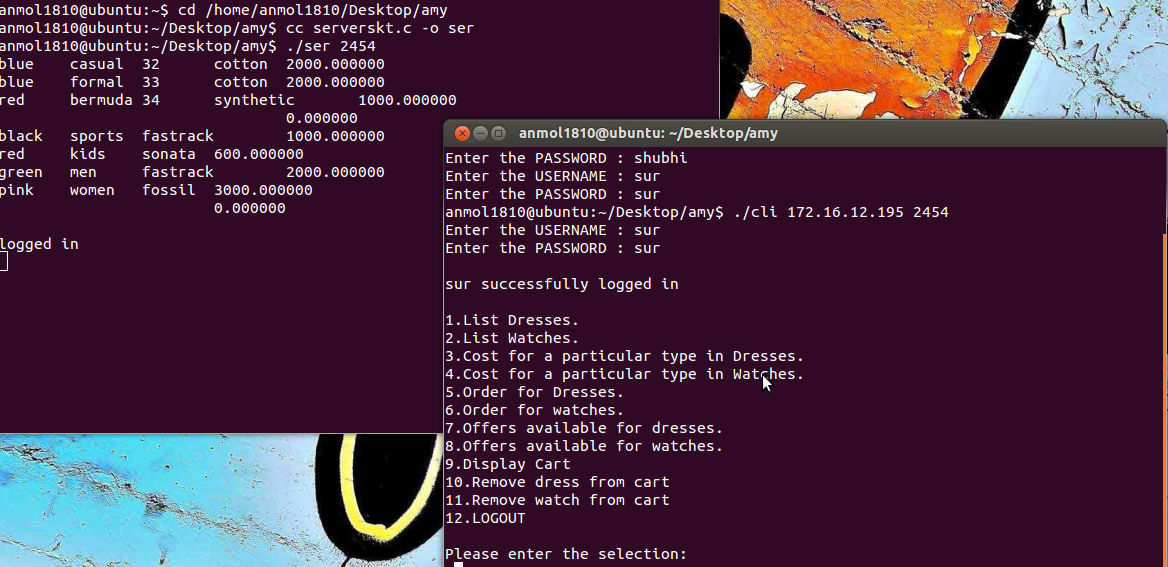
}

**Data Base Description:**

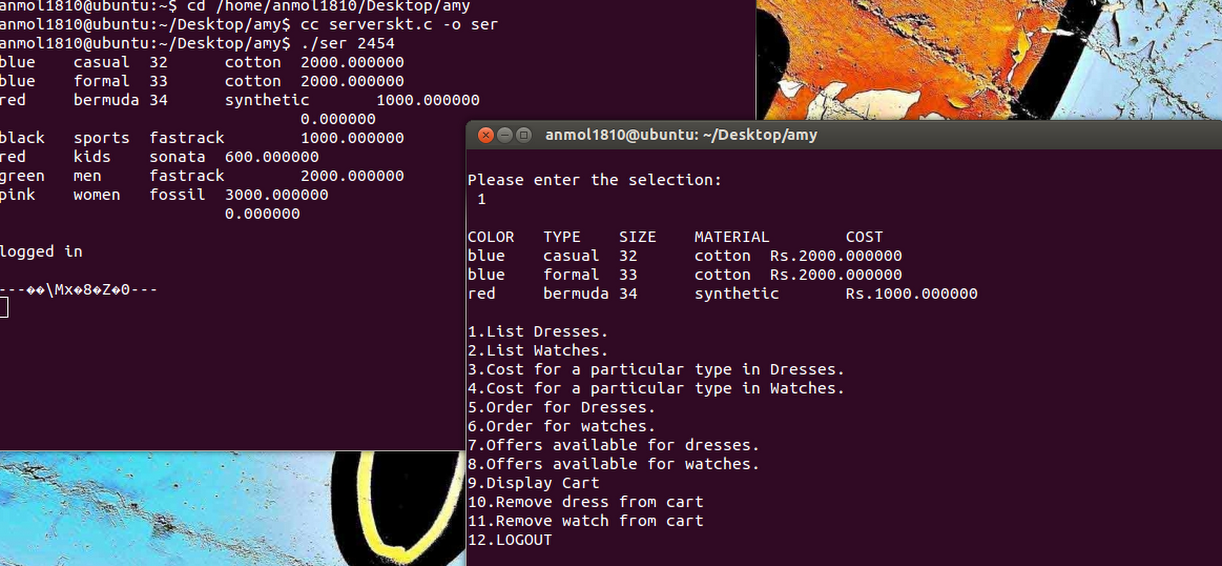
* **The first file used to store the customer details along with their order summary**
* **The other file used to store the dress details**
* **The other file is used store the watch details**
* **The new file is used to store the wish list for a customer for any items added to them in wish list**

**SCREENSHOTS:**

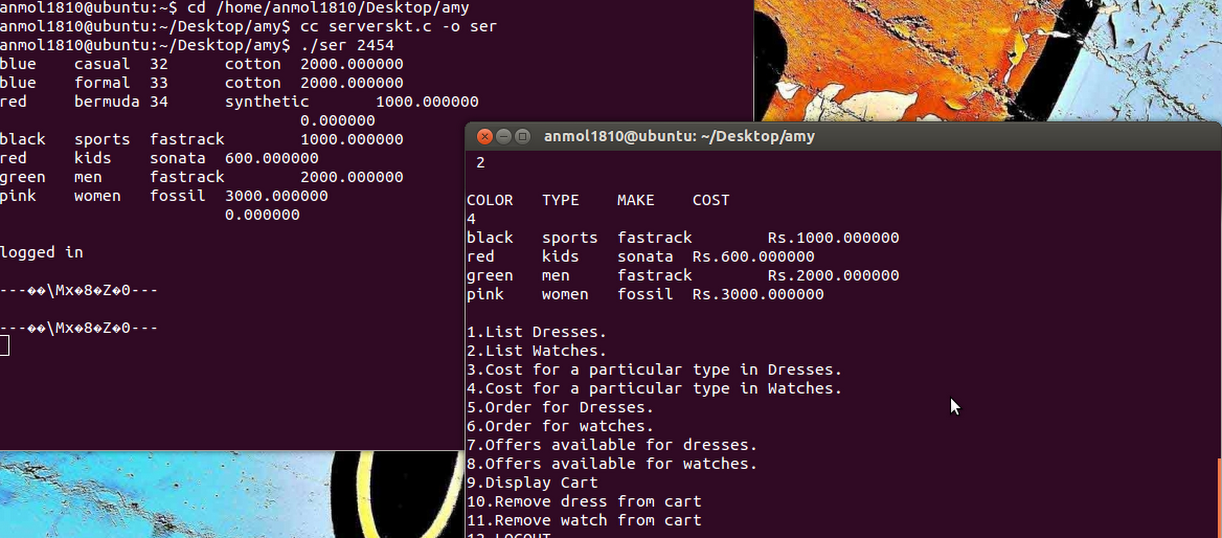
**Log in:**



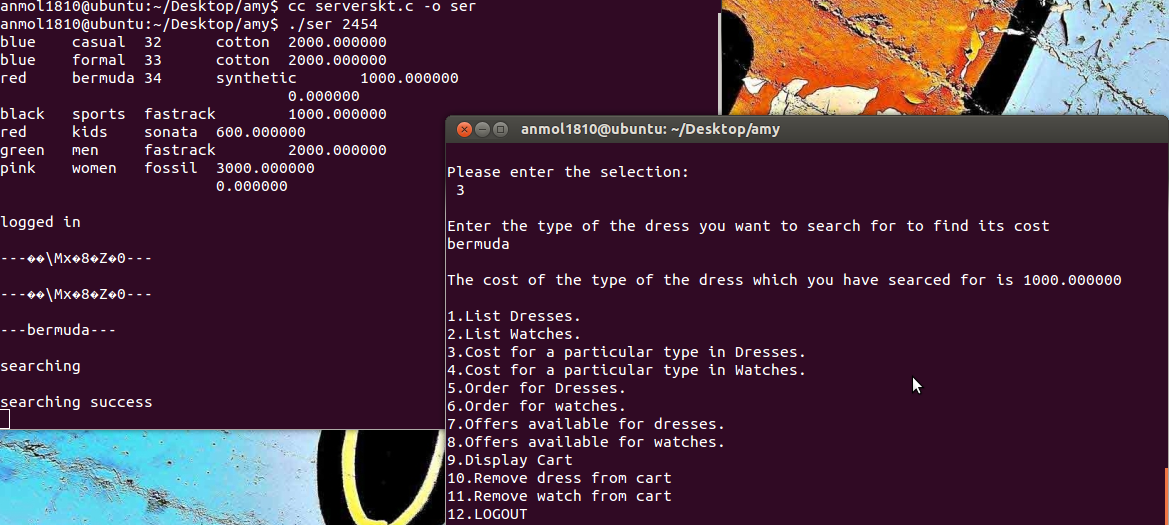
**Option 1: List Dresses**.



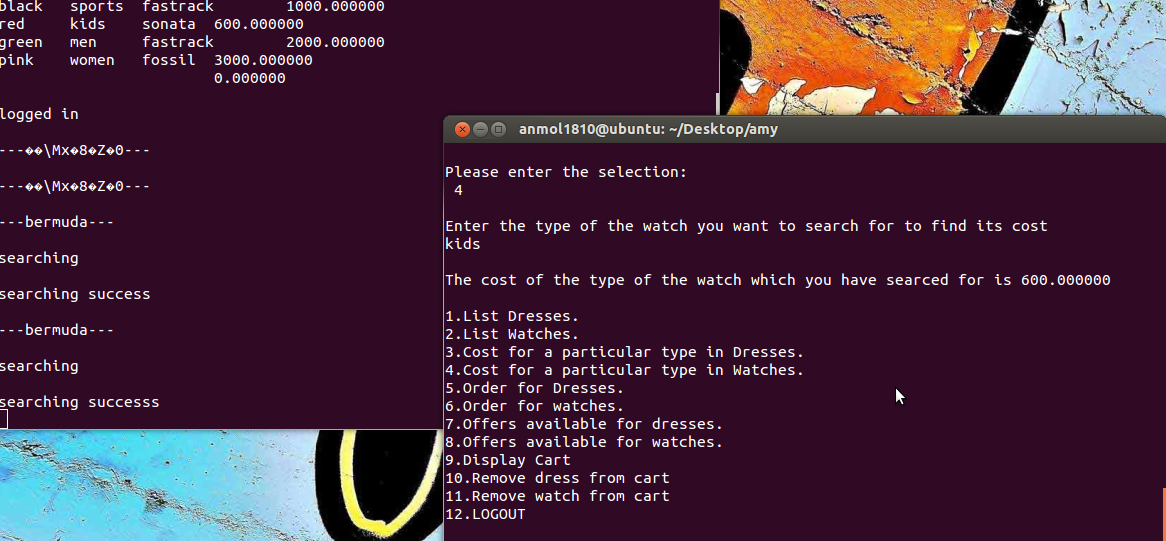
**Option 2:List Watches**.



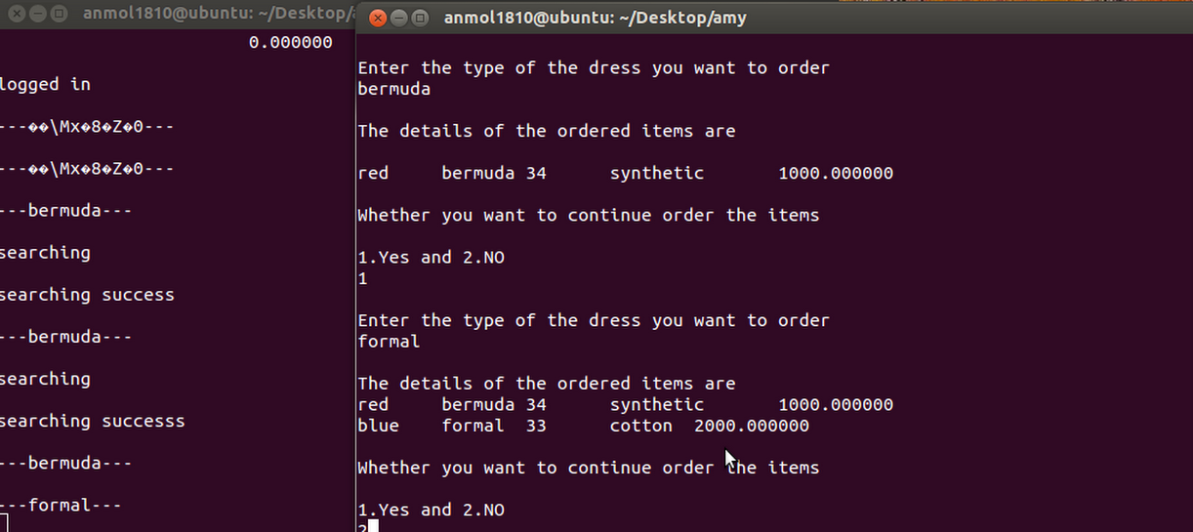
**Option 3:Cost for a particular type in Dresses**.



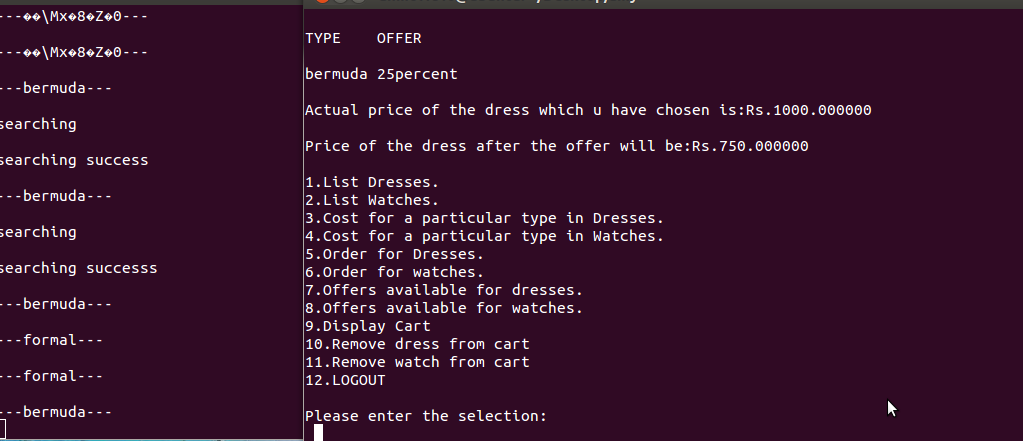
**Option 4:Cost for a particular type in Watches**.



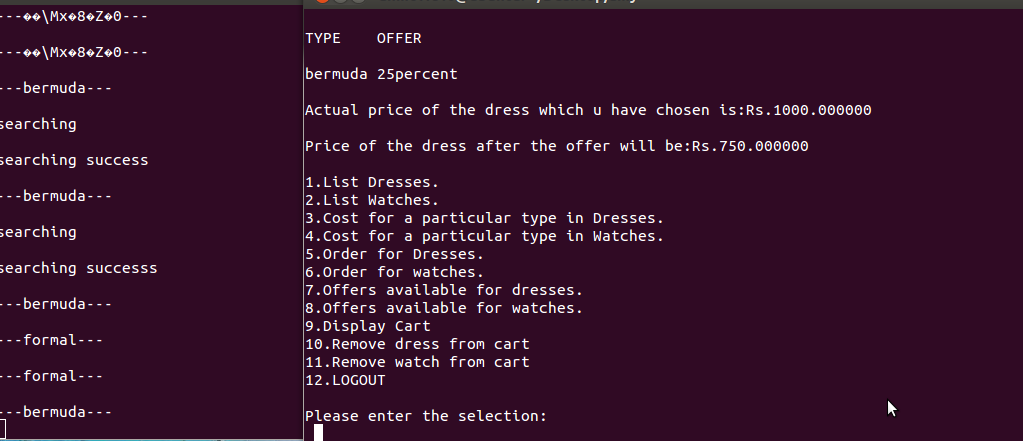
**Option 5:Order for Dresses**.



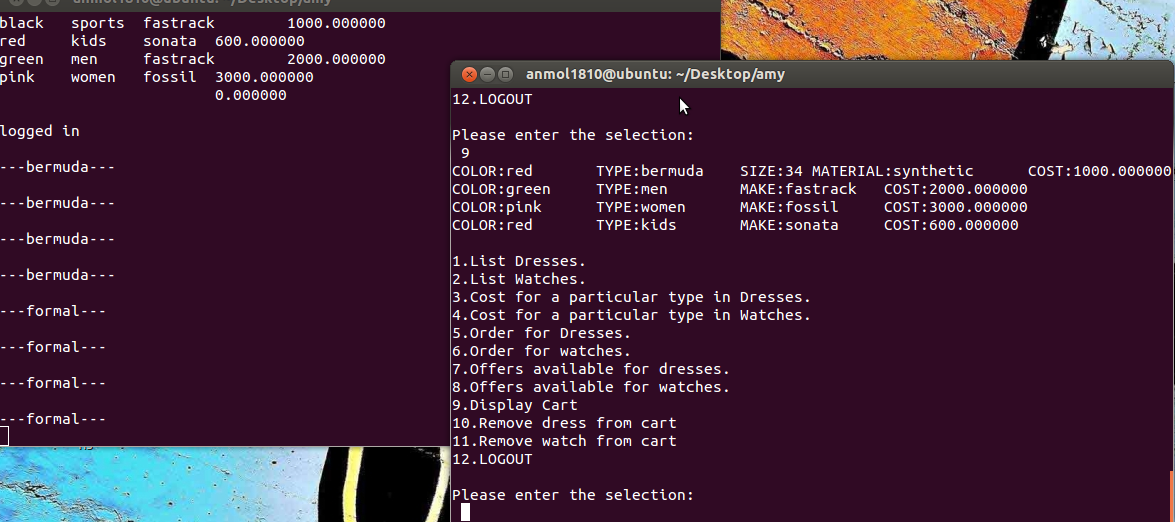
**Option 7:Offers available for Dresses**.



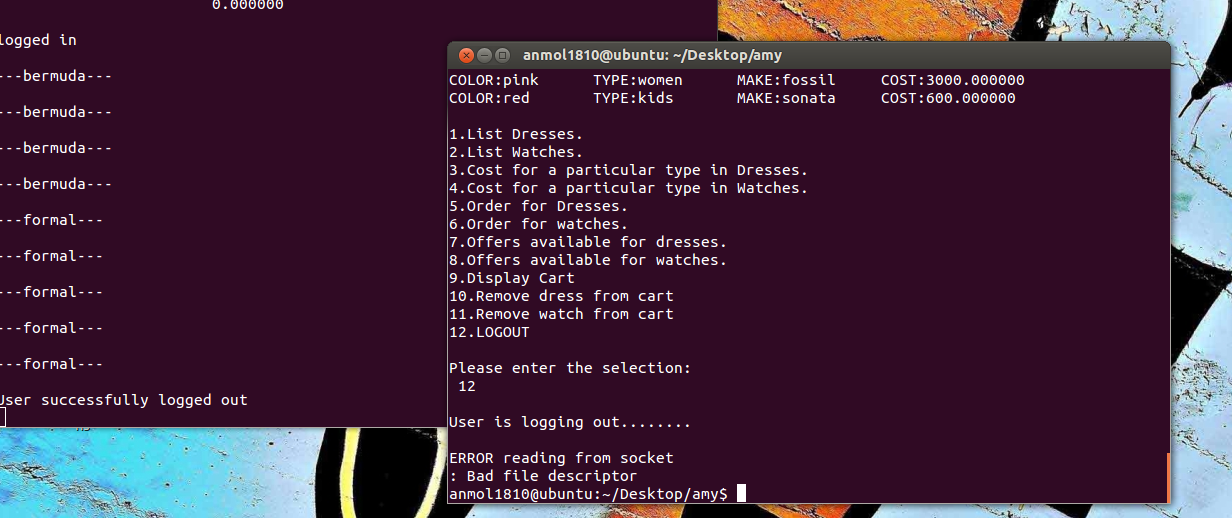
**Option 9:Display Cart**.



**Option 10:Remove dress from cart.**



**Option 12:LOGOUT**



**OBSERVATION:**

* The authentication is processed on the basis of the details entered by the client and if it is valid user the server allows the shopping for the dress or a watch
* The client-1 is connected to the server and successfully sends and receives the desired information to the server after performing the various search or placing the order
* The server acknowledges the result.
* The client receives the confirmation message if the order is placed or the wish list updating happens if the user adds any of the item to his/her wish list

**CONCLUSION:**

* The connection was successfully implemented between the clients and server.
* Communication between two clients and server was successfully established.
* Successfully implemented how the online shopping dress and watch portal.