OS MINI PROJECT -REPORT

***BY:** NAME:YASH GUPTA(Yash.Gupta514@iiitb.ac.in)*
ROLL NO:IMT2021514

Description:This is a server-client based project where the server and client communicates through socket programming.On client side there are two types of users:
1.Admin(super user) 2.Normal user
Admin can add,delete or update products.
Whereas,normal user can buy products,view its carts etc.

Total 7 files .
1.server.c-contains server side code
2.client.c-contains client side code
3.CUSTOMERS.txt
4.ORDERS.txt
5.REC.txt
6.receipt.txt-contains receipt of normal user
7.Areceipt.txt-contais all doing of admin

PROCEDURE TO RUN THE CODE:
use two terminals:

on 1st terminal:
type: gcc server.c -o server
on 2nd terminal:
type:gcc client.c -o client
then again on 1st terminal:
type;./server
again on second terminal:
type: ./client

Admin functions:
1.Add/Delete a product
2. Update the quantity/price of an existing product
3.Display all the products in the format: P_ID, P_Name, Cost

User functions:
1.Display all the products in the format: P_ID, P_Name, Cost
2.Add product to cart,it can buy multiple products
2.Display its Cart.
3. Edit its cart.
4.Payment of products in cart

<u>Functions on client side:</u>

sd-socket file descriptor of socket formed on client side

void loginadmin(int sd); WHEN admin logins,this function comes into play.This function gives menu to admin and performs functions of admin like adding a product,deleting or updating a product.

void loginuser(int sd);: When normal user login,this function comes into play.This function gives menu to user and performs functions like adding product to cart,viewing cart of user,editing cart of user,doing payment of items in cart etc.

void printProduct(struct Product p);-This function prints product list with their ids,names,prices and quantities

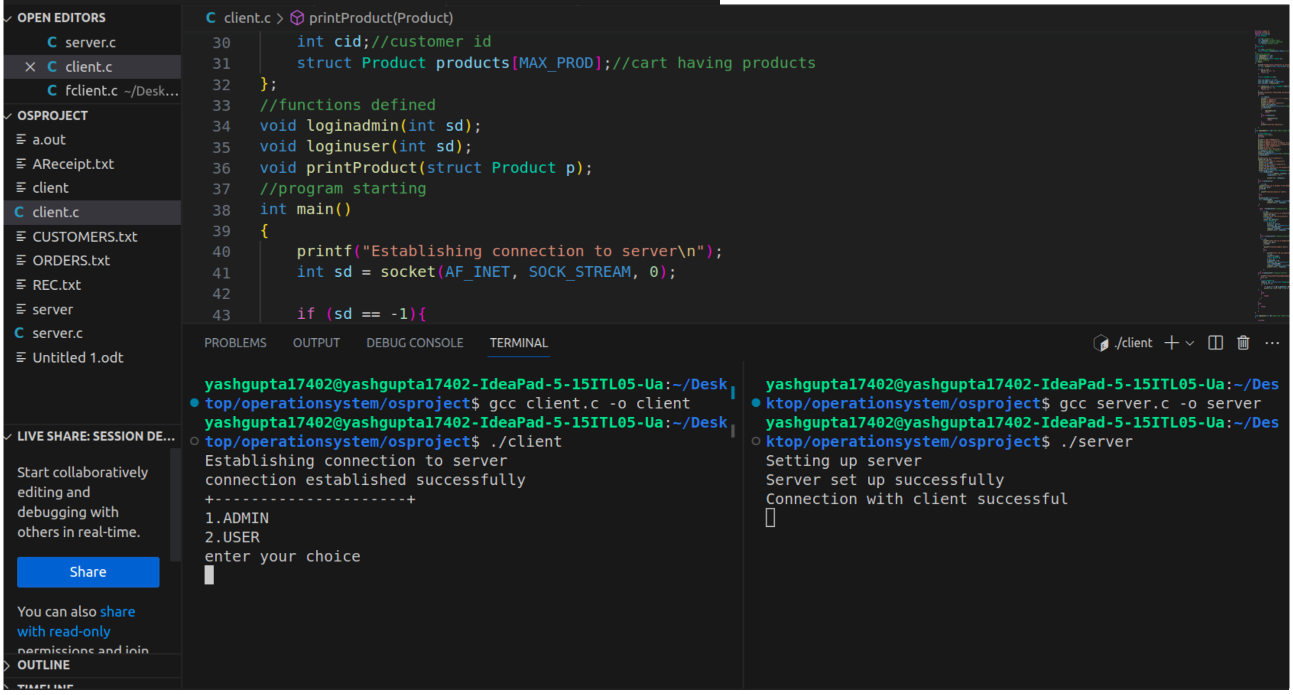
Functions on server side:

```
void setLockCust(int fd_custs, struct flock lock_cust);//setting locks on customers list
void unlock(int fd, struct flock lock);
void productReadLock(int fd, struct flock lock);//puts read lock on products
void productWriteLock(int fd, struct flock lock);//puts write lock on products
```

These functions puts locks on product ,customer list.

```
void listProducts(int fd, int new_fd);//lists the products
void addCustomer(int fd_cart, int fd_custs, int new_fd);//adding customner to customer list
void viewCart(int fd_cart, int new_fd, int fd_custs);//fcn to view cart for user
void addProductToCart(int fd, int fd_cart, int fd_custs, int new_fd);//adding product to the cart of user
void editProductInCart(int fd, int fd_cart, int fd_custs, int new_fd);//editing cart of user
void payment(int fd, int fd_cart, int fd_custs, int new_fd);//function perfroming payment for user
void generateAdminReceipt(int fd_admin, int fd);//function to generate admin receipt
void addProducts(int fd, int new_fd, int fd_admin);//function to add product
void deleteProduct(int fd, int new_fd, int id, int fd_admin);//function to delete product
void updateProduct(int fd, int new_fd, int ch, int fd_admin);//function to change price/quantity of product
```

WORKING OF CODE (SCREENSHOTS);

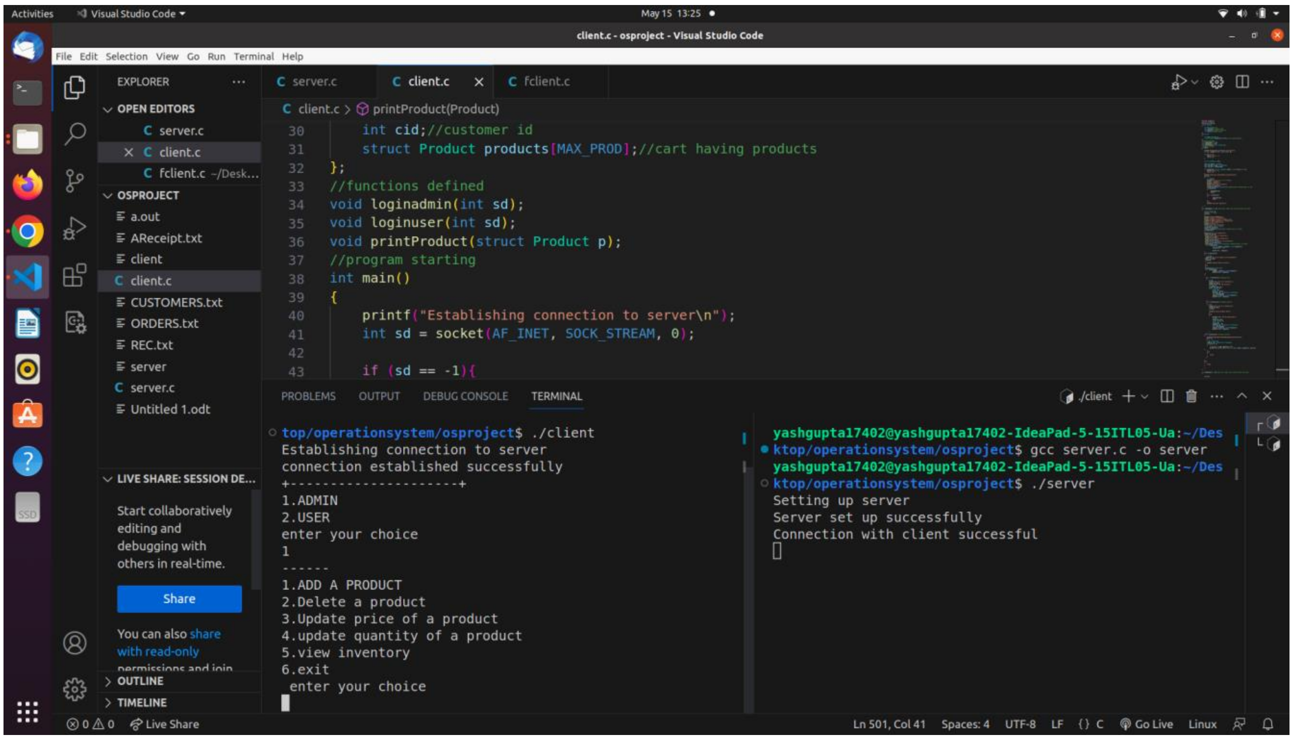


```
    int cid;//customer id
    struct Product products[MAX_PROD];//cart having products
};
//functions defined
void loginadmin(int sd);
void loginuser(int sd);
void printProduct(struct Product p);
//program starting
int main()
{
    printf("Establishing connection to server\n");
    int sd = socket(AF_INET, SOCK_STREAM, 0);

    if (sd == -1){
```

```
yashgupta17402@yashgupta17402-IdeaPad-5-15ITL05-Ua:~/Desk
top/operationsystem/osproject$ gcc client.c -o client
yashgupta17402@yashgupta17402-IdeaPad-5-15ITL05-Ua:~/Desk
top/operationsystem/osproject$ ./client
Establishing connection to server
connection established successfully
+---------------------+
1.ADMIN
2.USER
enter your choice
```

```
yashgupta17402@yashgupta17402-IdeaPad-5-15ITL05-Ua:~/Des
ktop/operationsystem/osproject$ gcc server.c -o server
yashgupta17402@yashgupta17402-IdeaPad-5-15ITL05-Ua:~/Des
ktop/operationsystem/osproject$ ./server
Setting up server
Server set up successfully
Connection with client successful
```

Admin's menu:

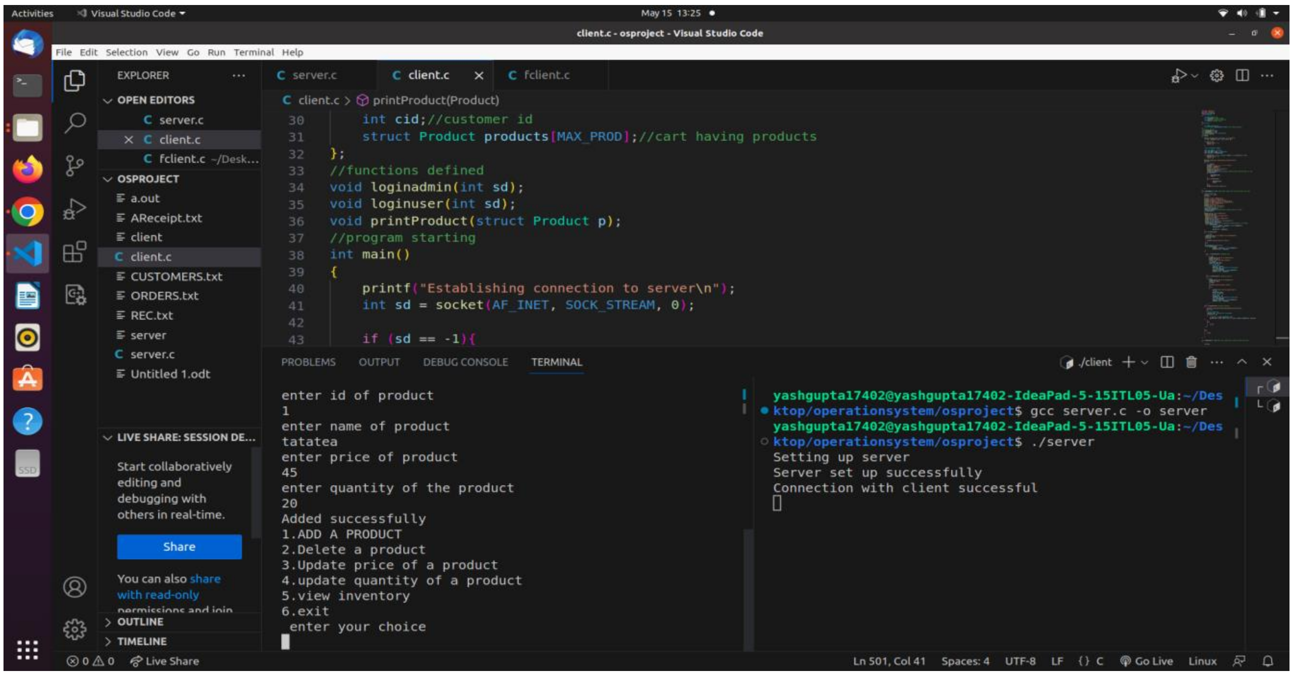


```
    int cid;//customer id
    struct Product products[MAX_PROD];//cart having products
};
//functions defined
void loginadmin(int sd);
void loginuser(int sd);
void printProduct(struct Product p);
//program starting
int main()
{
    printf("Establishing connection to server\n");
    int sd = socket(AF_INET, SOCK_STREAM, 0);

    if (sd == -1){
```

```
top/operationsystem/osproject$ ./client
Establishing connection to server
connection established successfully
+---------------------+
1.ADMIN
2.USER
enter your choice
1
------
1.ADD A PRODUCT
2.Delete a product
3.Update price of a product
4.update quantity of a product
5.view inventory
6.exit
 enter your choice
```

```
yashgupta17402@yashgupta17402-IdeaPad-5-15ITL05-Ua:~/Des
ktop/operationsystem/osproject$ gcc server.c -o server
yashgupta17402@yashgupta17402-IdeaPad-5-15ITL05-Ua:~/Des
ktop/operationsystem/osproject$ ./server
Setting up server
Server set up successfully
Connection with client successful
```

Adding of product by admin:

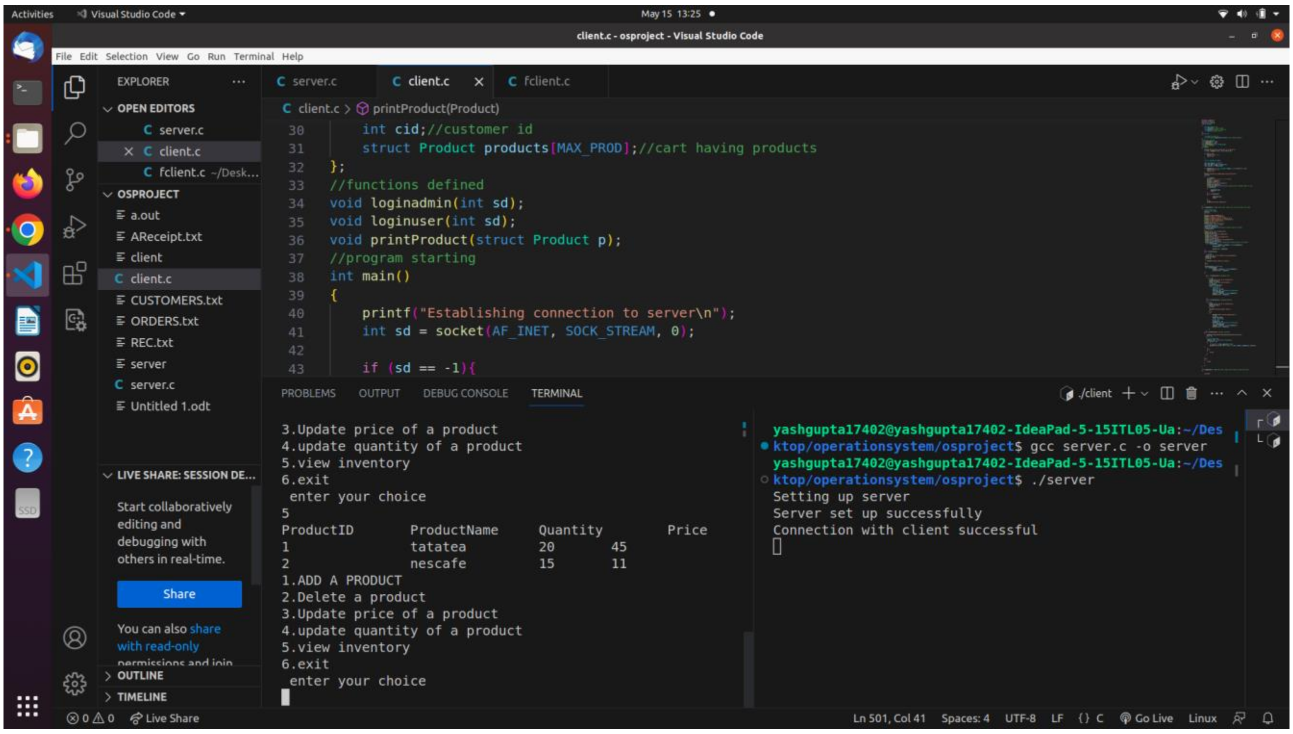


```
    int cid;//customer id
    struct Product products[MAX_PROD];//cart having products
};
//functions defined
void loginadmin(int sd);
void loginuser(int sd);
void printProduct(struct Product p);
//program starting
int main()
{
    printf("Establishing connection to server\n");
    int sd = socket(AF_INET, SOCK_STREAM, 0);

    if (sd == -1){
```

```
enter id of product
1
enter name of product
tatatea
enter price of product
45
enter quantity of the product
20
Added successfully
1.ADD A PRODUCT
2.Delete a product
3.Update price of a product
4.update quantity of a product
5.view inventory
6.exit
 enter your choice
```

```
yashgupta17402@yashgupta17402-IdeaPad-5-15ITL05-Ua:~/Desktop/operationsystem/osproject$ gcc server.c -o server
yashgupta17402@yashgupta17402-IdeaPad-5-15ITL05-Ua:~/Desktop/operationsystem/osproject$ ./server
Setting up server
Server set up successfully
Connection with client successful
```

Viewing inventory:

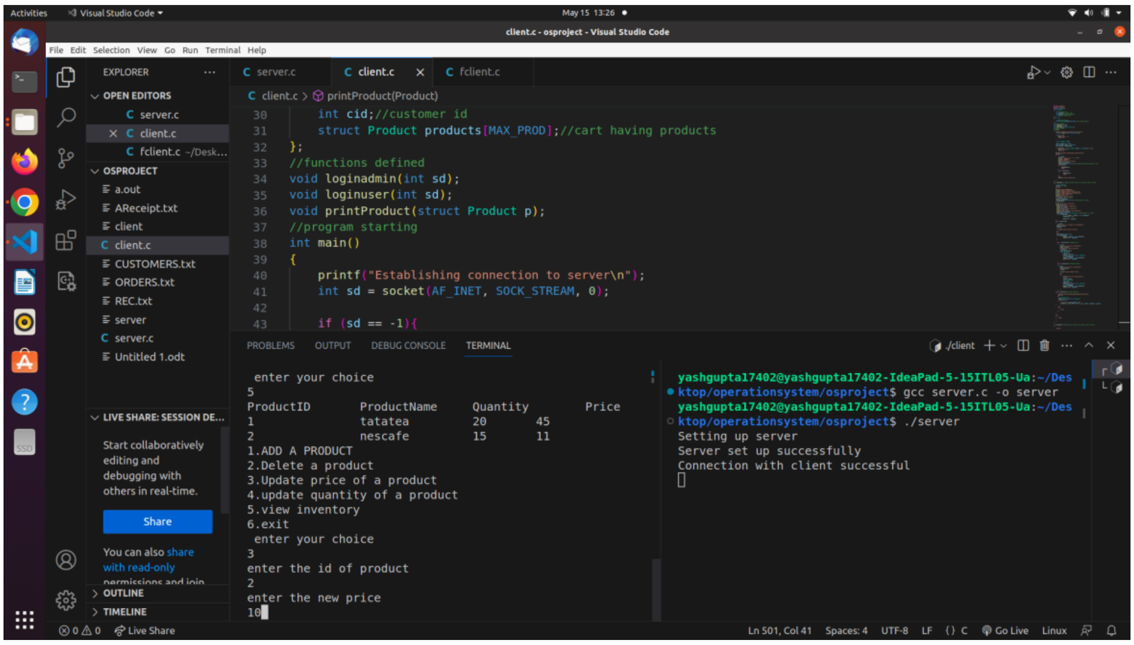


```
    int cid;//customer id
    struct Product products[MAX_PROD];//cart having products
};
//functions defined
void loginadmin(int sd);
void loginuser(int sd);
void printProduct(struct Product p);
//program starting
int main()
{
    printf("Establishing connection to server\n");
    int sd = socket(AF_INET, SOCK_STREAM, 0);

    if (sd == -1){
```

```
3.Update price of a product
4.update quantity of a product
5.view inventory
6.exit
 enter your choice
5
ProductID       ProductName     Quantity        Price
1               tatatea         20      45
2               nescafe         15      11
1.ADD A PRODUCT
2.Delete a product
3.Update price of a product
4.update quantity of a product
5.view inventory
6.exit
 enter your choice
```

```
yashgupta17402@yashgupta17402-IdeaPad-5-15ITL05-Ua:~/Desktop/operationsystem/osproject$ gcc server.c -o server
yashgupta17402@yashgupta17402-IdeaPad-5-15ITL05-Ua:~/Desktop/operationsystem/osproject$ ./server
Setting up server
Server set up successfully
Connection with client successful
```

Changing price of product with id=2

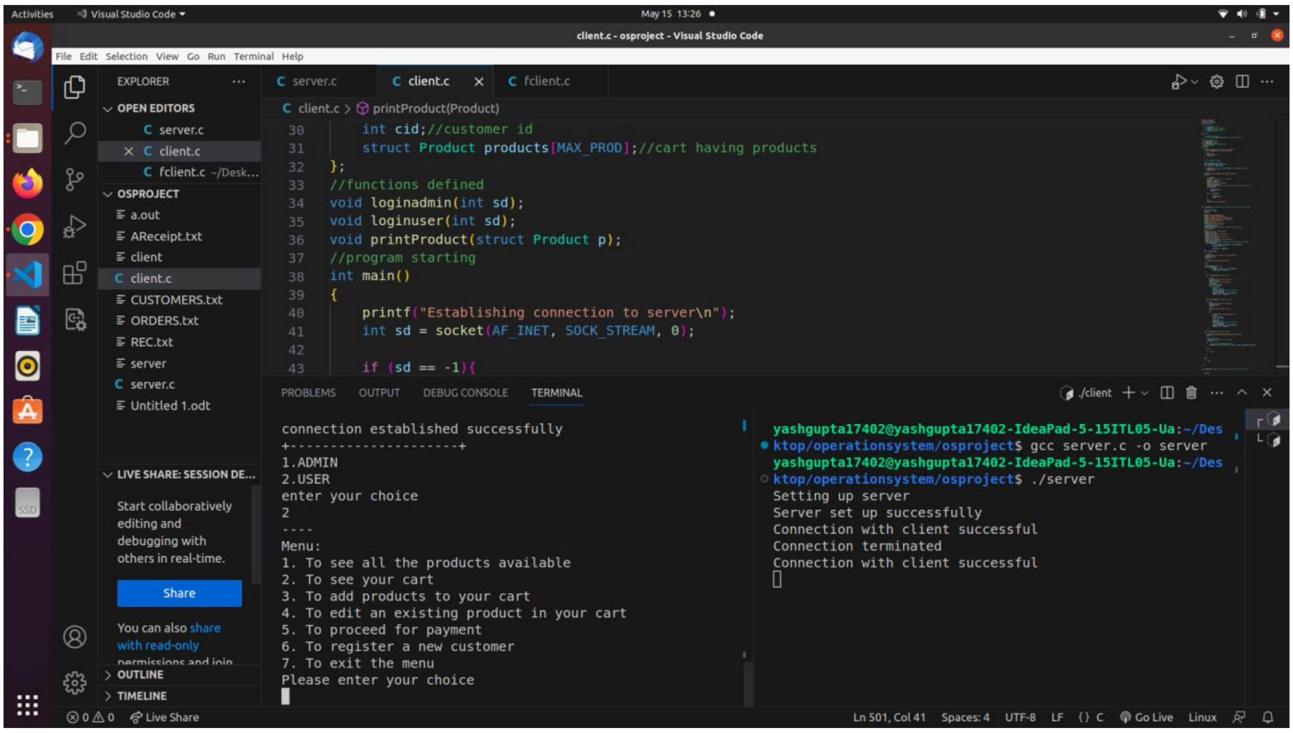


```c
    int cid;//customer id
    struct Product products[MAX_PROD];//cart having products
};
//functions defined
void loginadmin(int sd);
void loginuser(int sd);
void printProduct(struct Product p);
//program starting
int main()
{
    printf("Establishing connection to server\n");
    int sd = socket(AF_INET, SOCK_STREAM, 0);

    if (sd == -1){
```

```
 enter your choice
5
ProductID        ProductName      Quantity        Price
1                tatatea          20        45
2                nescafe          15        11
1.ADD A PRODUCT
2.Delete a product
3.Update price of a product
4.update quantity of a product
5.view inventory
6.exit
 enter your choice
3
enter the id of product
2
enter the new price
10
```

```
yashgupta17402@yashgupta17402-IdeaPad-5-15ITL05-Ua:~/Desktop/operationsystem/osproject$ gcc server.c -o server
yashgupta17402@yashgupta17402-IdeaPad-5-15ITL05-Ua:~/Desktop/operationsystem/osproject$ ./server
Setting up server
Server set up successfully
Connection with client successful
```

User's menu:

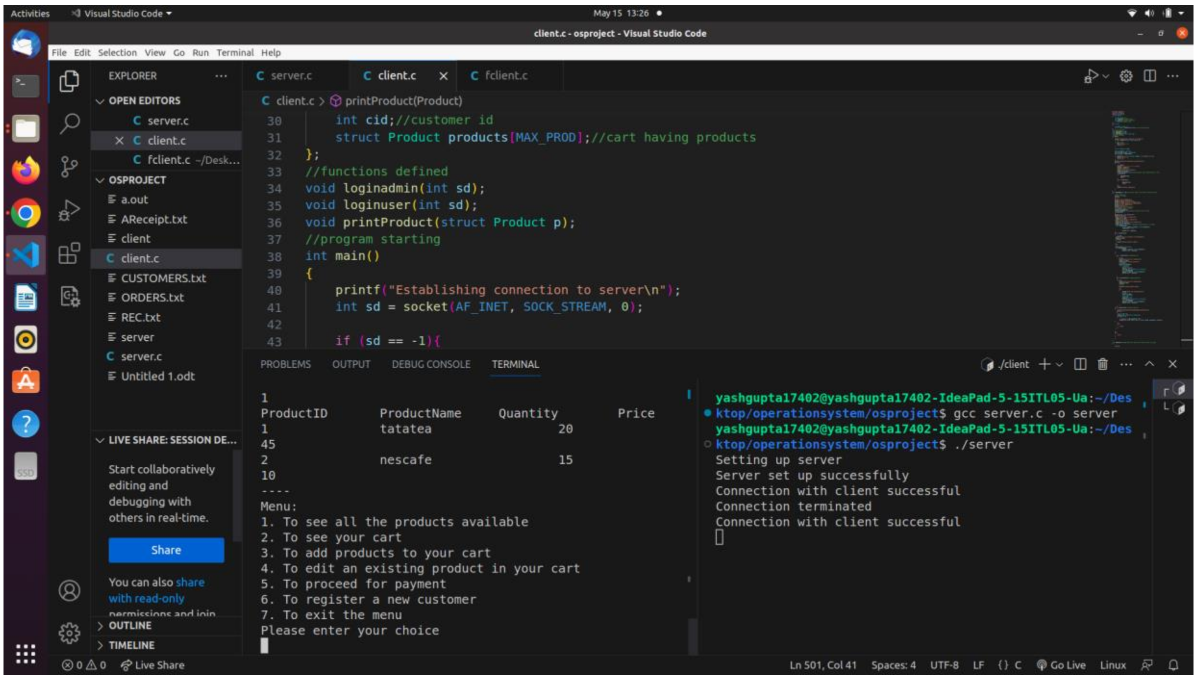


```c
    int cid;//customer id
    struct Product products[MAX_PROD];//cart having products
};
//functions defined
void loginadmin(int sd);
void loginuser(int sd);
void printProduct(struct Product p);
//program starting
int main()
{
    printf("Establishing connection to server\n");
    int sd = socket(AF_INET, SOCK_STREAM, 0);

    if (sd == -1){
```

```
connection established successfully
+----------------------+
1.ADMIN
2.USER
enter your choice
2
----
Menu:
1. To see all the products available
2. To see your cart
3. To add products to your cart
4. To edit an existing product in your cart
5. To proceed for payment
6. To register a new customer
7. To exit the menu
Please enter your choice
```

```
yashgupta17402@yashgupta17402-IdeaPad-5-15ITL05-Ua:~/Desktop/operationsystem/osproject$ gcc server.c -o server
yashgupta17402@yashgupta17402-IdeaPad-5-15ITL05-Ua:~/Desktop/operationsystem/osproject$ ./server
Setting up server
Server set up successfully
Connection with client successful
Connection terminated
Connection with client successful
```

Viewing products available:

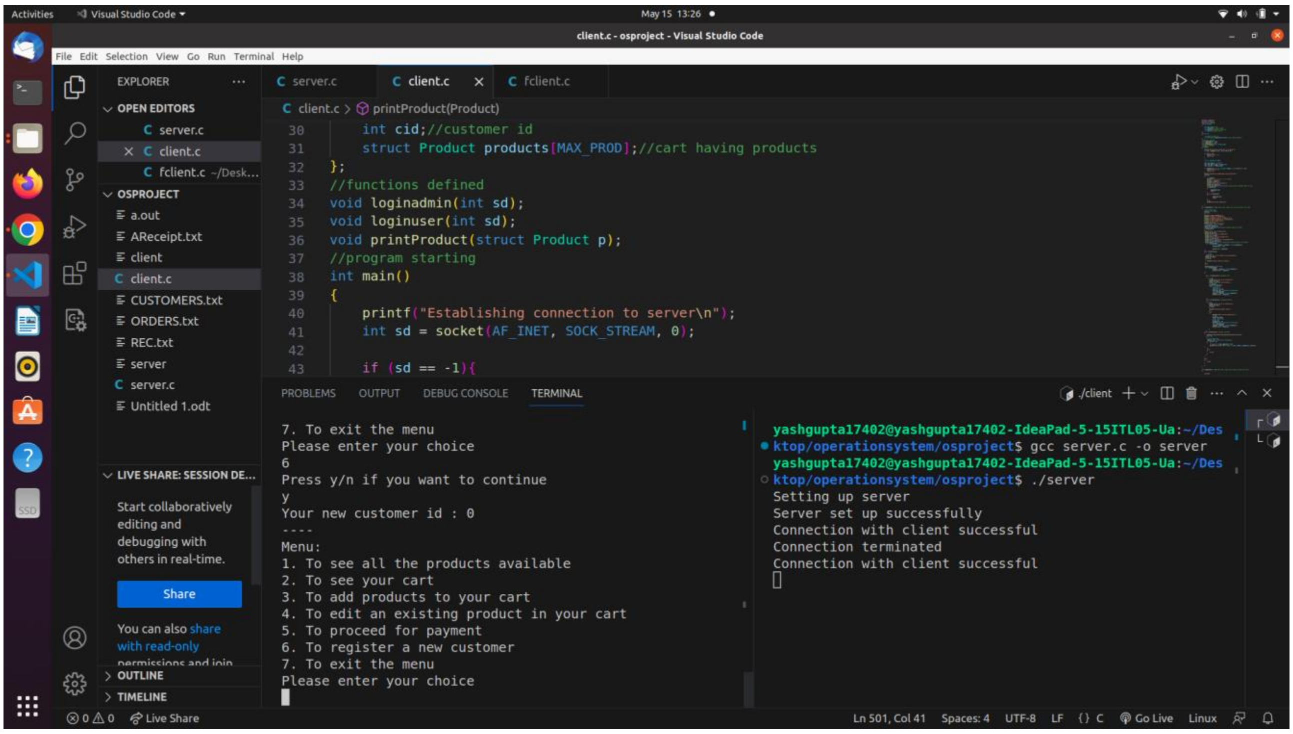


```
    int cid;//customer id
    struct Product products[MAX_PROD];//cart having products
};
//functions defined
void loginadmin(int sd);
void loginuser(int sd);
void printProduct(struct Product p);
//program starting
int main()
{
    printf("Establishing connection to server\n");
    int sd = socket(AF_INET, SOCK_STREAM, 0);

    if (sd == -1){
```

```
1
ProductID        ProductName      Quantity         Price
1                tatatea                20
45
2                nescafe                15
10
----
Menu:
1. To see all the products available
2. To see your cart
3. To add products to your cart
4. To edit an existing product in your cart
5. To proceed for payment
6. To register a new customer
7. To exit the menu
Please enter your choice
```

```
yashgupta17402@yashgupta17402-IdeaPad-5-15ITL05-Ua:~/Des
ktop/operationsystem/osproject$ gcc server.c -o server
yashgupta17402@yashgupta17402-IdeaPad-5-15ITL05-Ua:~/Des
ktop/operationsystem/osproject$ ./server
Setting up server
Server set up successfully
Connection with client successful
Connection terminated
Connection with client successful
```

Registering customer:

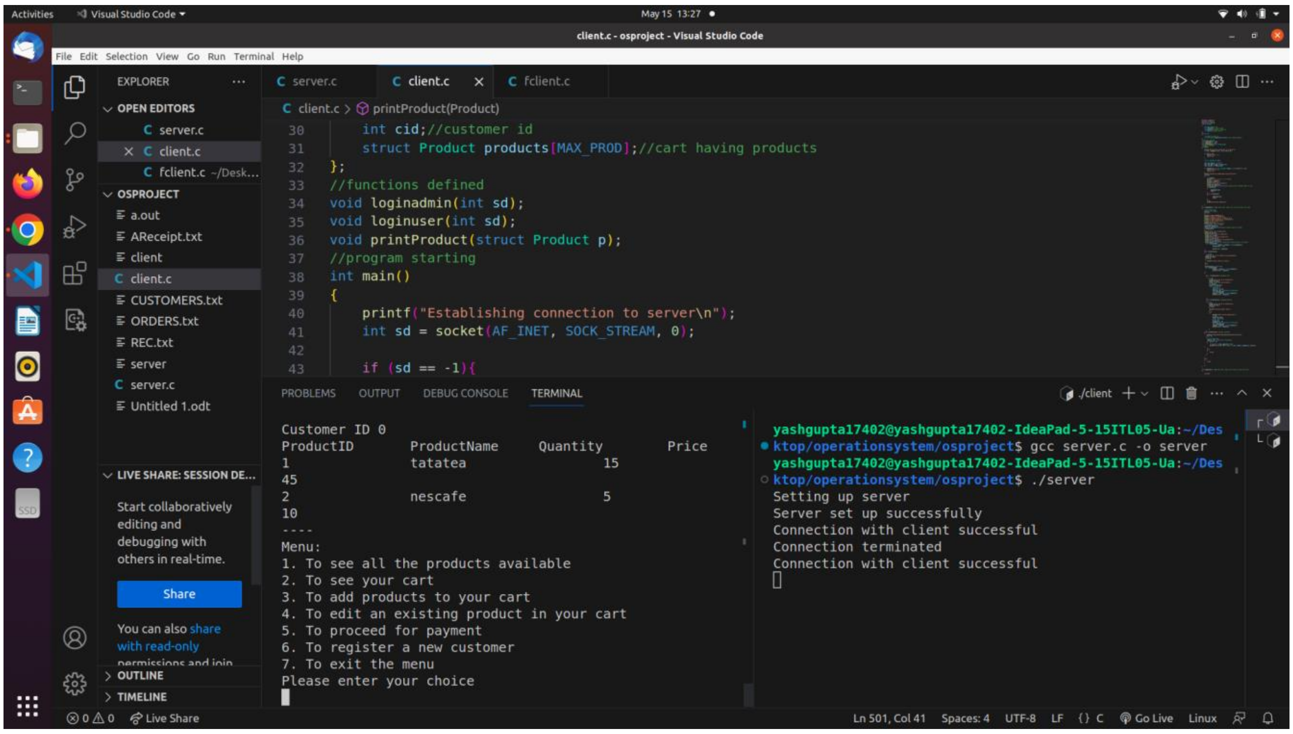


```
    int cid;//customer id
    struct Product products[MAX_PROD];//cart having products
};
//functions defined
void loginadmin(int sd);
void loginuser(int sd);
void printProduct(struct Product p);
//program starting
int main()
{
    printf("Establishing connection to server\n");
    int sd = socket(AF_INET, SOCK_STREAM, 0);

    if (sd == -1){
```

```
7. To exit the menu
Please enter your choice
6
Press y/n if you want to continue
y
Your new customer id : 0
----
Menu:
1. To see all the products available
2. To see your cart
3. To add products to your cart
4. To edit an existing product in your cart
5. To proceed for payment
6. To register a new customer
7. To exit the menu
Please enter your choice
```

```
yashgupta17402@yashgupta17402-IdeaPad-5-15ITL05-Ua:~/Des
ktop/operationsystem/osproject$ gcc server.c -o server
yashgupta17402@yashgupta17402-IdeaPad-5-15ITL05-Ua:~/Des
ktop/operationsystem/osproject$ ./server
Setting up server
Server set up successfully
Connection with client successful
Connection terminated
Connection with client successful
```

Viewing cart of customer

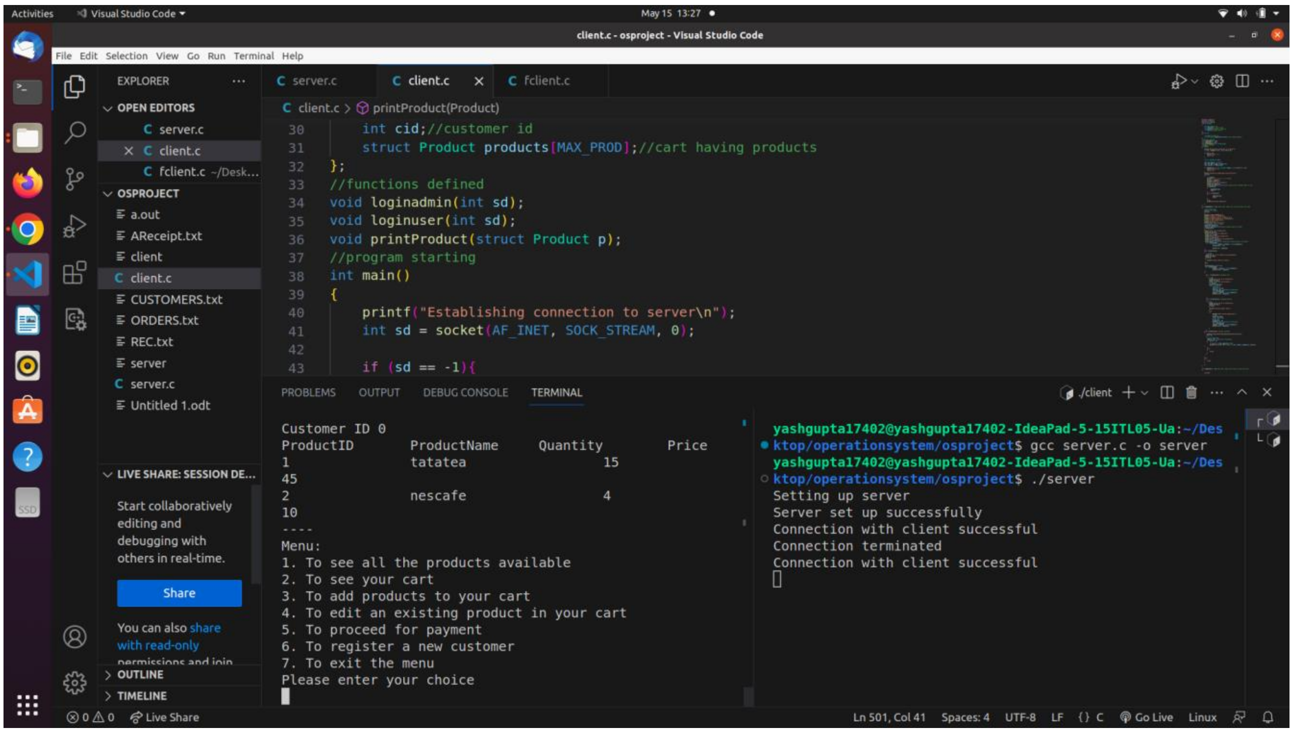


```c
    int cid;//customer id
    struct Product products[MAX_PROD];//cart having products
};
//functions defined
void loginadmin(int sd);
void loginuser(int sd);
void printProduct(struct Product p);
//program starting
int main()
{
    printf("Establishing connection to server\n");
    int sd = socket(AF_INET, SOCK_STREAM, 0);

    if (sd == -1){
```

```
Customer ID 0
ProductID        ProductName      Quantity         Price
1                tatatea                15
45
2                nescafe                5
10
----
Menu:
1. To see all the products available
2. To see your cart
3. To add products to your cart
4. To edit an existing product in your cart
5. To proceed for payment
6. To register a new customer
7. To exit the menu
Please enter your choice
```

```
yashgupta17402@yashgupta17402-IdeaPad-5-15ITL05-Ua:~/Des
ktop/operationsystem/osproject$ gcc server.c -o server
yashgupta17402@yashgupta17402-IdeaPad-5-15ITL05-Ua:~/Des
ktop/operationsystem/osproject$ ./server
Setting up server
Server set up successfully
Connection with client successful
Connection terminated
Connection with client successful
```

Edited quantity of product (id=2)

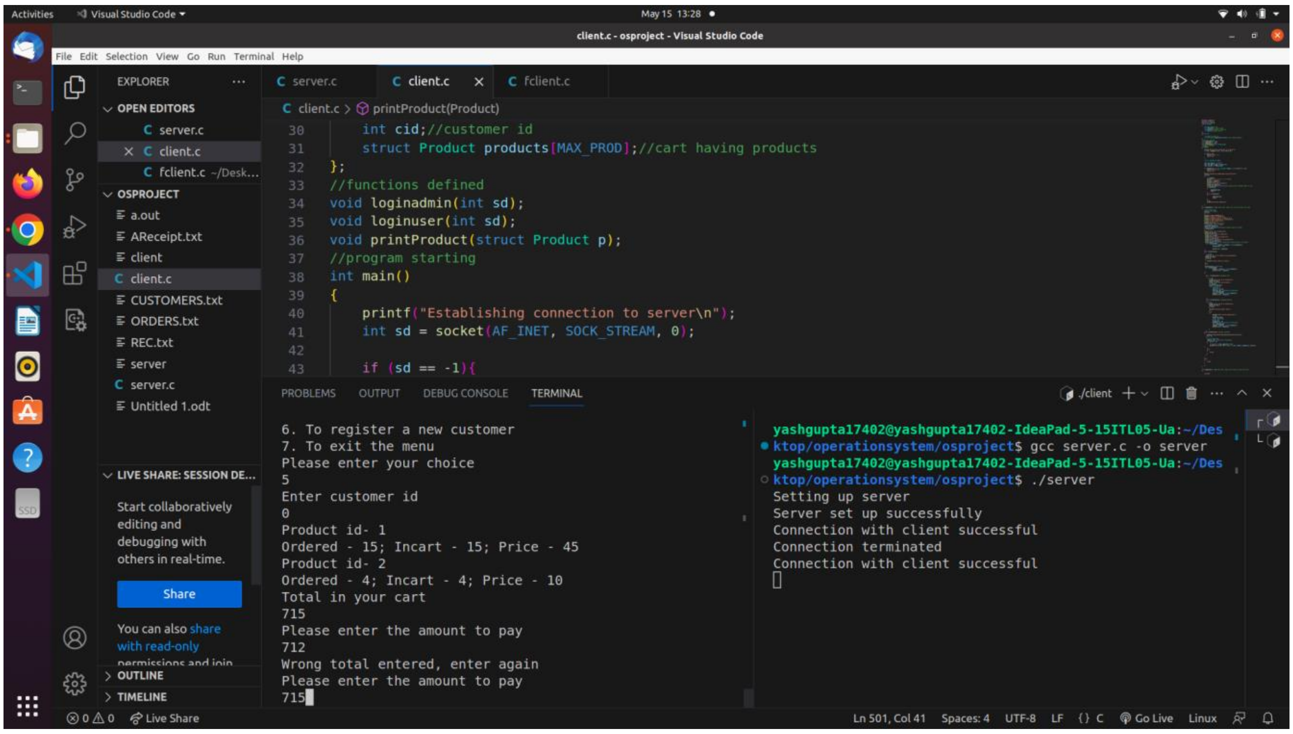


```c
    int cid;//customer id
    struct Product products[MAX_PROD];//cart having products
};
//functions defined
void loginadmin(int sd);
void loginuser(int sd);
void printProduct(struct Product p);
//program starting
int main()
{
    printf("Establishing connection to server\n");
    int sd = socket(AF_INET, SOCK_STREAM, 0);

    if (sd == -1){
```

```
Customer ID 0
ProductID        ProductName      Quantity         Price
1                tatatea                15
45
2                nescafe                4
10
----
Menu:
1. To see all the products available
2. To see your cart
3. To add products to your cart
4. To edit an existing product in your cart
5. To proceed for payment
6. To register a new customer
7. To exit the menu
Please enter your choice
```

```
yashgupta17402@yashgupta17402-IdeaPad-5-15ITL05-Ua:~/Des
ktop/operationsystem/osproject$ gcc server.c -o server
yashgupta17402@yashgupta17402-IdeaPad-5-15ITL05-Ua:~/Des
ktop/operationsystem/osproject$ ./server
Setting up server
Server set up successfully
Connection with client successful
Connection terminated
Connection with client successful
```

Payment:

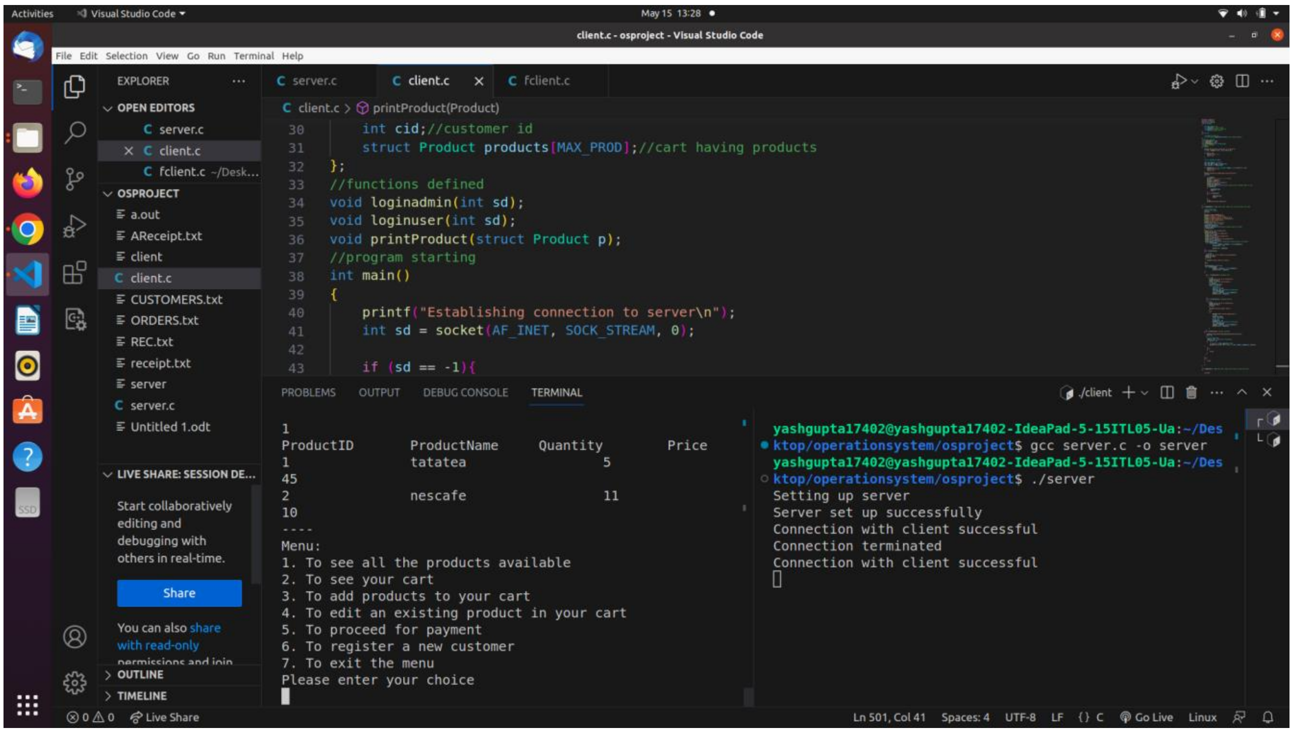

Activities
Visual Studio Code
May 15 13:28
client.c - osproject - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
OPEN EDITORS
server.c
client.c
fclient.c ~/Desk...
OSPROJECT
a.out
AReceipt.txt
client
client.c
CUSTOMERS.txt
ORDERS.txt
REC.txt
server
server.c
Untitled 1.odt
LIVE SHARE: SESSION DE...
Start collaboratively editing and debugging with others in real-time.
Share
You can also share with read-only permissions and join
OUTLINE
TIMELINE
client.c > printProduct(Product)
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
./client
Ln 501, Col 41 Spaces: 4 UTF-8 LF {} C Go Live Linux
Live Share

```c
    int cid;//customer id
    struct Product products[MAX_PROD];//cart having products
};
//functions defined
void loginadmin(int sd);
void loginuser(int sd);
void printProduct(struct Product p);
//program starting
int main()
{
    printf("Establishing connection to server\n");
    int sd = socket(AF_INET, SOCK_STREAM, 0);

    if (sd == -1){
```

```
6. To register a new customer
7. To exit the menu
Please enter your choice
5
Enter customer id
0
Product id- 1
Ordered - 15; Incart - 15; Price - 45
Product id- 2
Ordered - 4; Incart - 4; Price - 10
Total in your cart
715
Please enter the amount to pay
712
Wrong total entered, enter again
Please enter the amount to pay
715
```

```
yashgupta17402@yashgupta17402-IdeaPad-5-15ITL05-Ua:~/Desktop/operationsystem/osproject$ gcc server.c -o server
yashgupta17402@yashgupta17402-IdeaPad-5-15ITL05-Ua:~/Desktop/operationsystem/osproject$ ./server
Setting up server
Server set up successfully
Connection with client successful
Connection terminated
Connection with client successful
```

Products available left :

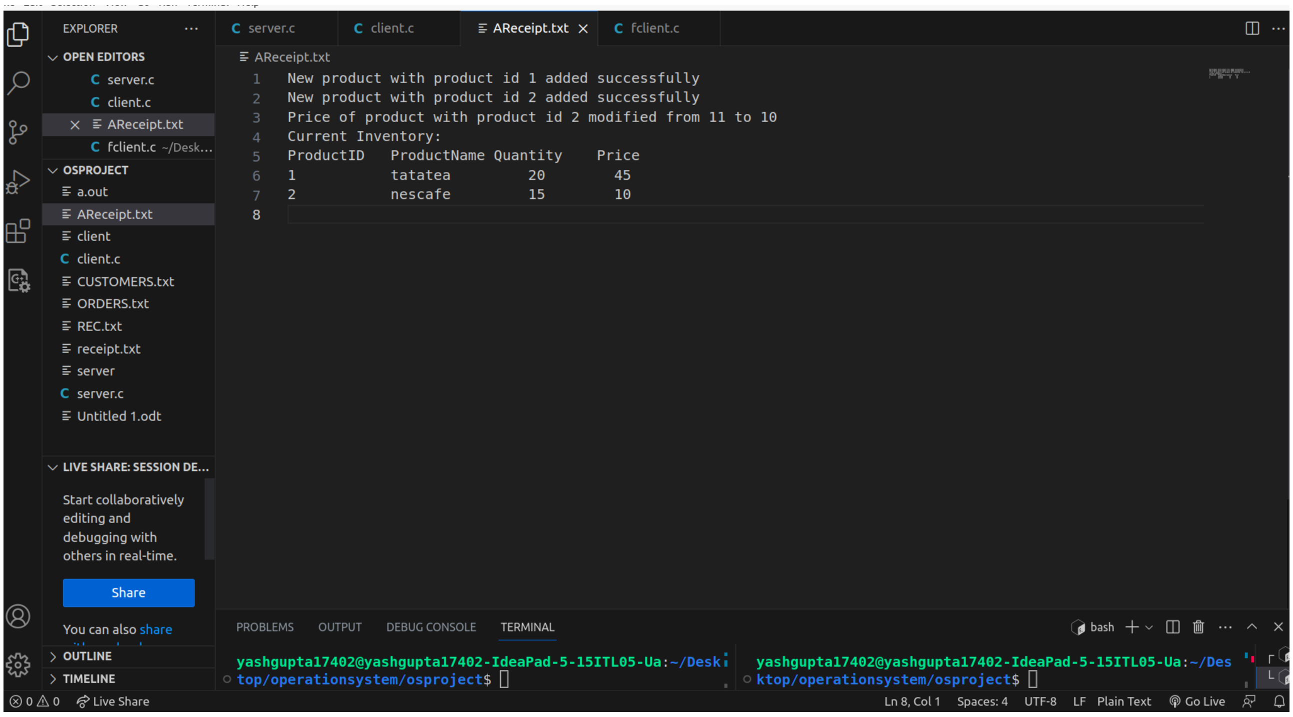

Activities
Visual Studio Code
May 15 13:28
client.c - osproject - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
OPEN EDITORS
server.c
client.c
fclient.c ~/Desk...
OSPROJECT
a.out
AReceipt.txt
client
client.c
CUSTOMERS.txt
ORDERS.txt
REC.txt
receipt.txt
server
server.c
Untitled 1.odt
LIVE SHARE: SESSION DE...
Start collaboratively editing and debugging with others in real-time.
Share
You can also share with read-only permissions and join
OUTLINE
TIMELINE
client.c > printProduct(Product)
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
./client
Ln 501, Col 41 Spaces: 4 UTF-8 LF {} C Go Live Linux
Live Share

```c
    int cid;//customer id
    struct Product products[MAX_PROD];//cart having products
};
//functions defined
void loginadmin(int sd);
void loginuser(int sd);
void printProduct(struct Product p);
//program starting
int main()
{
    printf("Establishing connection to server\n");
    int sd = socket(AF_INET, SOCK_STREAM, 0);

    if (sd == -1){
```

```
1
ProductID       ProductName     Quantity        Price
1               tatatea                 5
45
2               nescafe                 11
10
----
Menu:
1. To see all the products available
2. To see your cart
3. To add products to your cart
4. To edit an existing product in your cart
5. To proceed for payment
6. To register a new customer
7. To exit the menu
Please enter your choice
```

```
yashgupta17402@yashgupta17402-IdeaPad-5-15ITL05-Ua:~/Desktop/operationsystem/osproject$ gcc server.c -o server
yashgupta17402@yashgupta17402-IdeaPad-5-15ITL05-Ua:~/Desktop/operationsystem/osproject$ ./server
Setting up server
Server set up successfully
Connection with client successful
Connection terminated
Connection with client successful
```

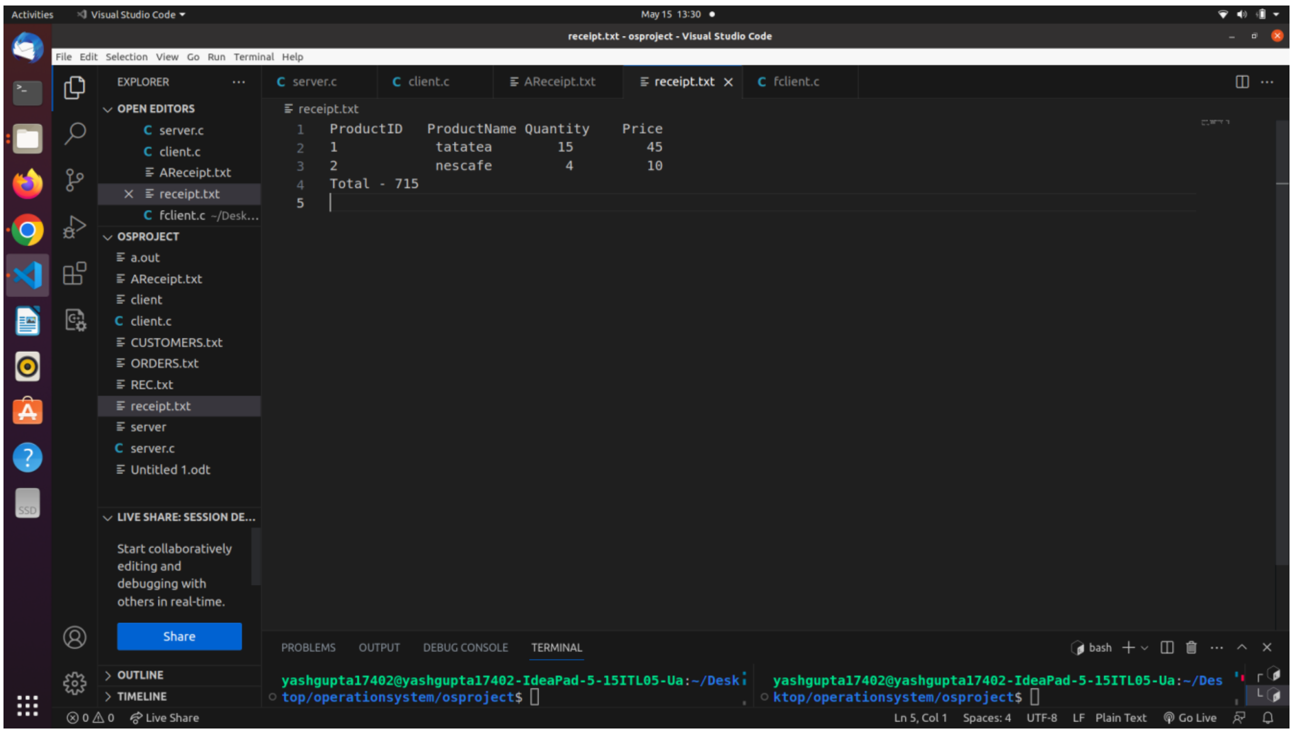
Admin's receipt



```
New product with product id 1 added successfully
New product with product id 2 added successfully
Price of product with product id 2 modified from 11 to 10
Current Inventory:
ProductID   ProductName Quantity    Price
1           tatatea         20        45
2           nescafe         15        10
```

```
yashgupta17402@yashgupta17402-IdeaPad-5-15ITL05-Ua:~/Desktop/operationsystem/osproject$
```

Order's receipt



```
ProductID   ProductName Quantity    Price
1            tatatea         15        45
2            nescafe         4         10
Total - 715
```

```
yashgupta17402@yashgupta17402-IdeaPad-5-15ITL05-Ua:~/Desktop/operationsystem/osproject$
```

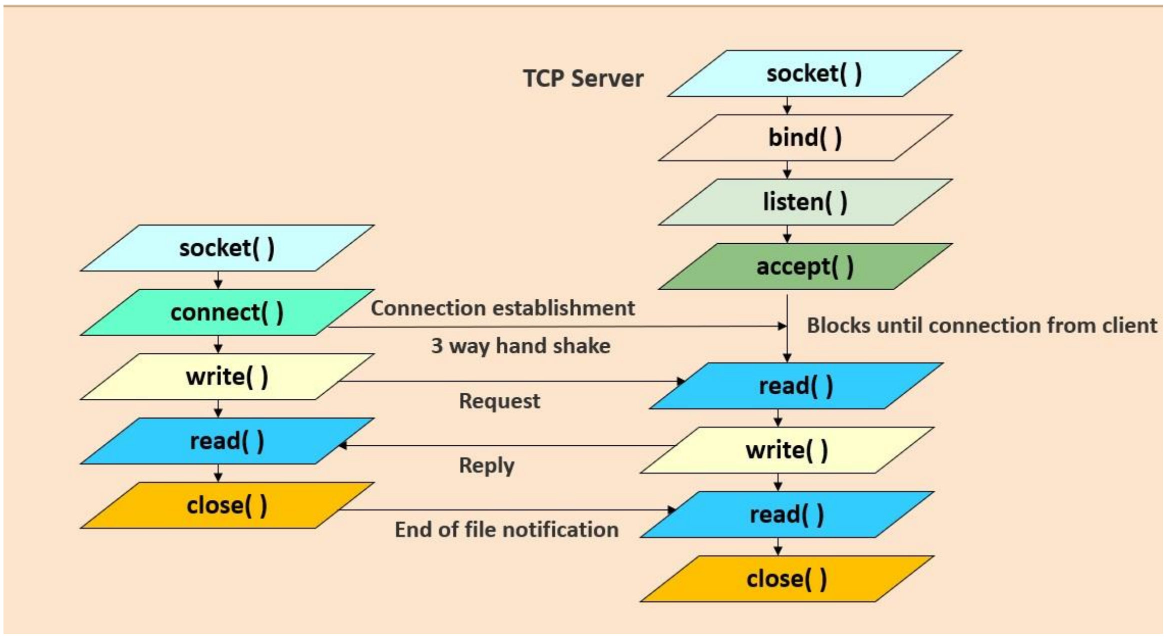
CONCEPTS USED:

1.SOCKET PROGRAMMING:

Socket programming is a way of writing networked applications in which different processes or computers can communicate with each other using the Internet Protocol (IP).

In socket programming, a server process creates a socket and binds it to a specific IP address and port number on the machine. This socket is then set up to listen for incoming connections from client processes. When a client process wants to connect to the server, it creates its own socket and specifies the server's IP address and port number. The server can then accept the connection and create a new socket to handle the communication with that client.

Once a connection is established between a client and server, they can communicate by sending data through their sockets. This data can be in any format, but it is typically sent in small chunks, or packets, to avoid overwhelming the network or the receiving process.

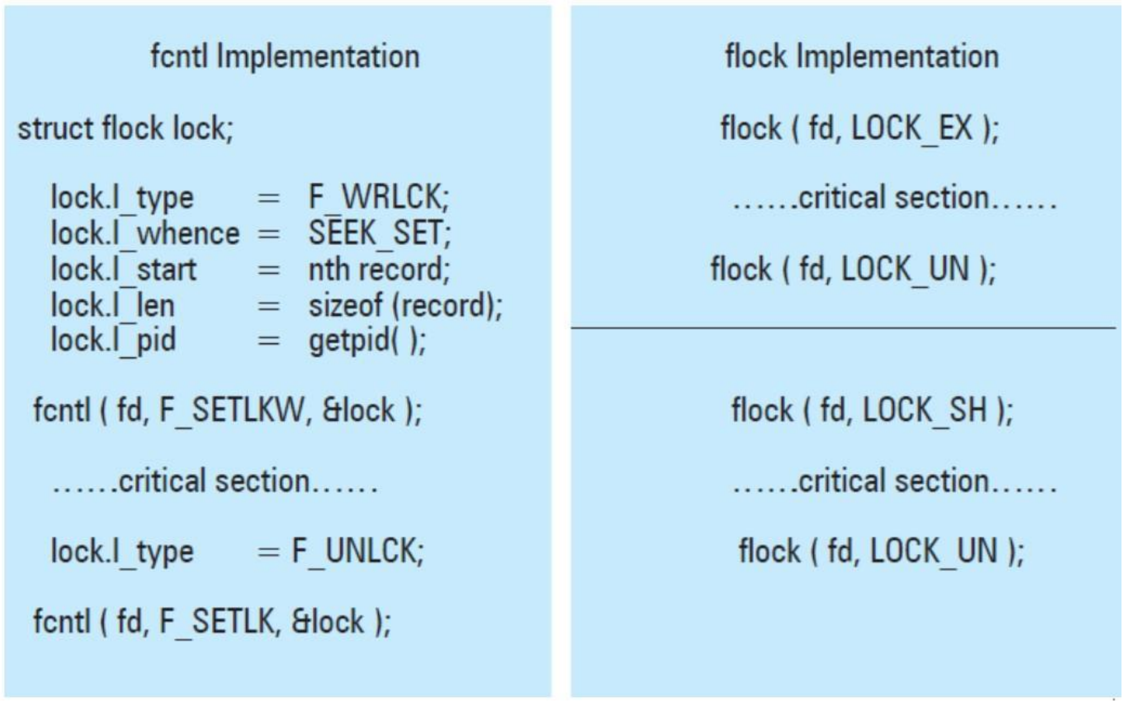


2.File Locking:

Mandatory and record-locking techniques are used to manage concurrent access to the data

.fcntl() is used.

When we try to find the customer id in the customers.txt file, we perform a mandatory read lock. Read lock is also used while displaying the inventory. The record write lock is used while adding a new product and deleting a product. The record write lock is used while adding a new customer. Write lock is used in the payment gateway.



fcntl Implementation

```
struct flock lock;

   lock.l_type      =  F_WRLCK;
   lock.l_whence    =  SEEK_SET;
   lock.l_start     =  nth record;
   lock.l_len       =  sizeof (record);
   lock.l_pid       =  getpid( );

 fcntl ( fd, F_SETLKW, &lock );

   ......critical section......

   lock.l_type      = F_UNLCK;

 fcntl ( fd, F_SETLK, &lock );
```

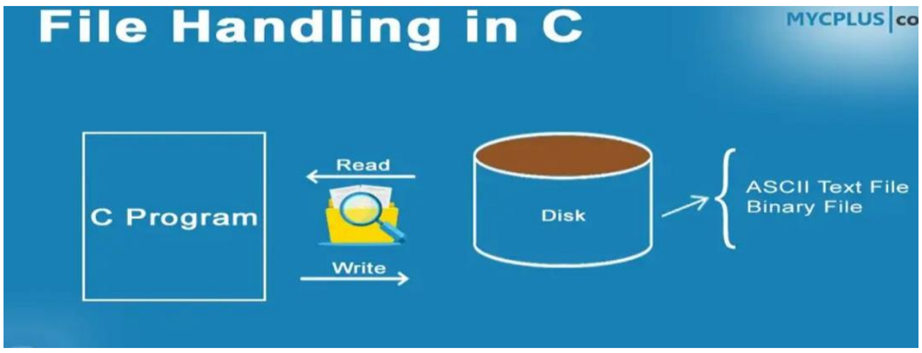
flock Implementation

```
flock ( fd, LOCK_EX );

  ......critical section......

flock ( fd, LOCK_UN );

flock ( fd, LOCK_SH );

  ......critical section......

flock ( fd, LOCK_UN );
```

3.File Handling:

Instead of setting up a database, files have been used to store and access data for the ease of implementation.

Hence to store, read and write to the files, file handling is used. Functions like open(),read(),write() are used.


File Handling in C
MYCPLUS|co
C Program
Read
Write
Disk
ASCII Text File
Binary File