Mini Project - OnlineStore

Karan Raj Pandey
IMT2021014

This project is terminal based and a simulation of an Online Store that uses a client-server architecture. The client sends requests to the server through sockets, which processes the request and sends back the response. The users are classified as normal users and admin users, and each user has their privileges. A normal user can view their cart, while an admin user has special privileges to update the store items. The communication between the client and server is done through system calls, and the server interacts with the data files using file locking mechanisms.

The application contains following files:

• Client.c - contains the code that handles client side operations.
• Server.c - contains the code that handles server side operations.
• cust.txt - contains the customers registered in the store's system.
• Orderlist.txt - contains the carts for each of the customer.
• rec.txt - contains the products currently available in the store with their quantities and prices, i.e the inventory of the store.
• receipt.txt - contains the receipt of the customer (or user) after payment.
• Receipt_Admin.txt - this file is used to store the history of the updates, deletes and additions of the products being dine by the admin.
• headers.h - this file contains the structs and macros to be used in the program.

This application offers different functionalities to two types of users - **Normal Users (Customers)** and **Admin Users**:

Normal Users can register themselves as new customers to the store, view products available in the store with their respective prices and quantities, add items to their cart, edit quantities or remove items from their cart, make payment, and view the receipt for their order.

On the other hand, **Admin Users** have the authority to manage the products in the store, including adding new products, updating the quantity or price of existing products, removing products from the store, and viewing the inventory. Any update made by the Admin User will be recorded in the Receipt_Admin.txt file.

Implementation of Server functions

The server program handles input sent by clients regarding user types and choices, and calls appropriate helper functions based on the case. For user functions, the program checks if the user wants to exit the program, view inventory, view cart, add/edit a product in the cart, pay for their cart, or add a new customer. The corresponding functions, such as listProducts(), viewCart(), addProductToCart(), editProductToCart(), payment(), and addCustomer() are called accordingly.

On the other hand, for admin functions, the program checks if the admin wants to add a product to the store, delete a product from the inventory, edit the price/quantity of a product in the inventory, view inventory, or exit the program. The corresponding functions, such as addProducts(), deleteProduct(), updateProduct(), and listProducts() are called accordingly, with the appropriate parameters passed in case of updateProduct().

Implementation of Client.c

The provided server code for an online store includes several helper functions to modularize the code and improve its readability. These functions are used for file locking, getting offsets, adding products to the

inventory, listing products, deleting products, updating products, adding customers, viewing carts, adding products to carts, editing products in carts, and processing payments.

The file locking functions are used to access the data files safely and include setLockCust(), productReadLock(), productWriteLock(), CartOffsetLock(), and unlock(). These functions ensure that the appropriate portions of the file are locked and unlocked whenever file access is required.

The getOffset() function is used internally by many other functions in the server code to find the cart offset for a given customer ID in the cust.txt file. This offset is used whenever a user's cart needs to be updated.

The addProducts() function is used by the admin to add products to the inventory. It checks if the product ID is already present and adds the product if it is not. A log statement is written to the Receipt_Admin.txt file to indicate the addition of the product.

The listProducts() function displays the products in the inventory and can be used by both admin and user.

The deleteProduct() function is used by the admin to delete a product from the inventory. It checks if the product ID is valid and deletes the product if it is. A log statement is written to the Receipt_Admin.txt file to indicate the deletion of the product.

The updateProduct() function updates the price or quantity of a product in the inventory. If the quantity is 0, the deleteProduct() function is called. A log statement is written to the Receipt_Admin.txt file to indicate the update.

The addCustomer() function adds a new customer with an auto-generated customer ID.

The viewCart() function takes a customer ID as an argument and uses the getOffset() function to find the cart offset. If the cart offset is valid, the function writes the corresponding cart to the client.

The addProductToCart() function adds a product to a customer's cart after performing various checks, including checking the customer ID, product ID, quantity requested, and stock availability. The function also checks the MAX PROD limit for the number of products that can be in a cart.

The editProductInCart() function allows a user to change the quantity of an item they have already ordered. The function performs various checks, including checking the customer ID, product ID, and new quantity. The product ID is also checked for its presence in the cart. If everything is correct, the product is updated in the customer's cart, or an error message is displayed.

The payment() function processes the payment for the products in the customer's cart. The function checks if the requested quantity is still in the cart and adjusts the total based on the current stock. The user inputs the amount displayed on the screen, and all changes are recorded in the inventory. The customer's cart is cleared, and the items bought are reflected in the receipt.txt file as a receipt.

The generateAdminReceipt() function is called when the admin exits the program. It writes the updated inventory to the Receipt_Admin.txt file to indicate all the changes made by the admin.

OS concepts used:

Socket programming

The application uses sockets to communicate between the server and the client.
1. The program uses the server side socket system calls (socket, bind, listen, accept) to setup the server side socket, and client side socket system calls (socket, connect) to setup the client side socket.
2. The fork() system call is used for setting up a concurrent server.
3. All reads and writes (from the socket) use read and write system calls.

File handling

1. Use of basic functions such as read, write and open to perform the basic write and read operations on the files.

File Locking

The program uses mandatory as well as record locking in the following places in the program:
1. On trying to find the customer id in the cust.txt file, we put a mandatory read lock on the entire file.
2. On finding the offset of the cart for the customer, we do a record locking at that offset in Orderlist.txt to lock that cart for reading or writing.
3. On trying to find the product id in the list of products, we put a mandatory read lock on all products. This is also used when inventory is displayed.
4. On updating a particular product, we remove the read lock on the products and acquire a write lock on that product, to change it.

Steps to run the Application

OnlineStore
An online store implemented using system calls and Linux programming in C as a part of the Operating Systems course.

Client Menu :
You can login as one of the following:
1. User (has permissions only limited to the items in his cart, to add items, remove items and to pay for his cart)
2. Admin (has permissions to update the products and add products to the inventory, and also to delete products)
User (has permissions only limited to the items in his cart, to add items, remove items and to pay for his cart)

Admin Functionalities:
1. Add/Delete a product
2. Update the quantity of a new product, or update the price

User Functionalities:
1. See the list of products the server has.
2. See his own cart.
3. Add/delete items from his cart
4. Update quantities in his cart
5. Pay for the items in his cart, and generate a receipt for the same.

Program Architecture
The program's architecture involves two separate programs - Server.c and Client.c. Server.c functions as the server code, while Client.c functions as the client code. Users can log in as either a regular user or an admin using the client code.

To facilitate communication between the server and client, the program utilizes sockets. Additionally, file locking is employed when accessing the program's data files, which include cust.txt (a list of registered customers), Orderlist.txt (the shopping cart for each customer), rec.txt (a record of the program's inventory), and receipt.txt (a record of completed payments).

Therefore, the program's architecture relies on two distinct programs, socket communication, and file locking to manage customer data and the program's inventory.

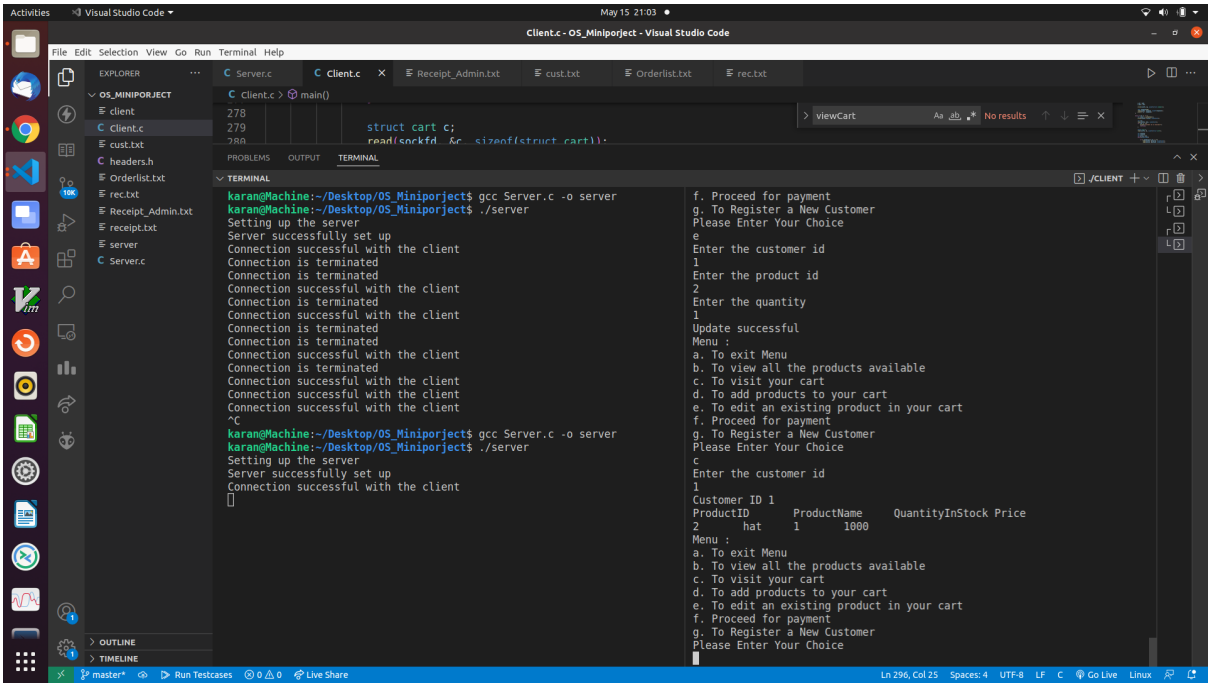
Instructions to run the program
Open a terminal, and run the following commands

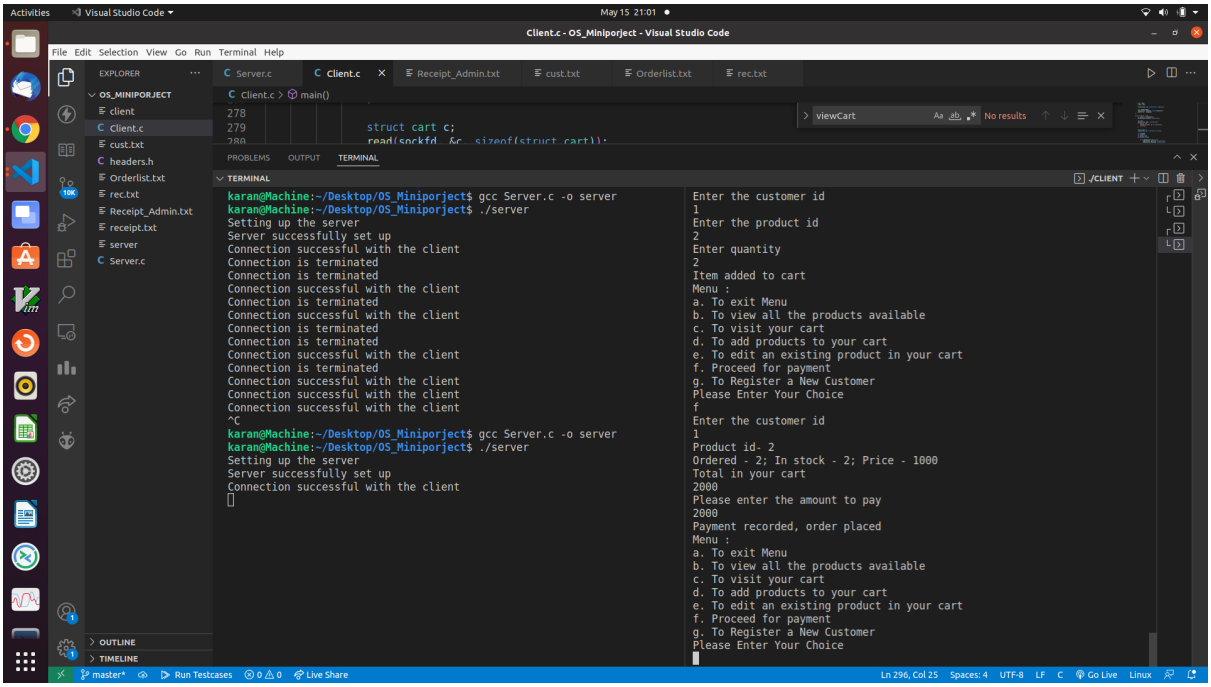
```
gcc -o server Server.c
./server
```

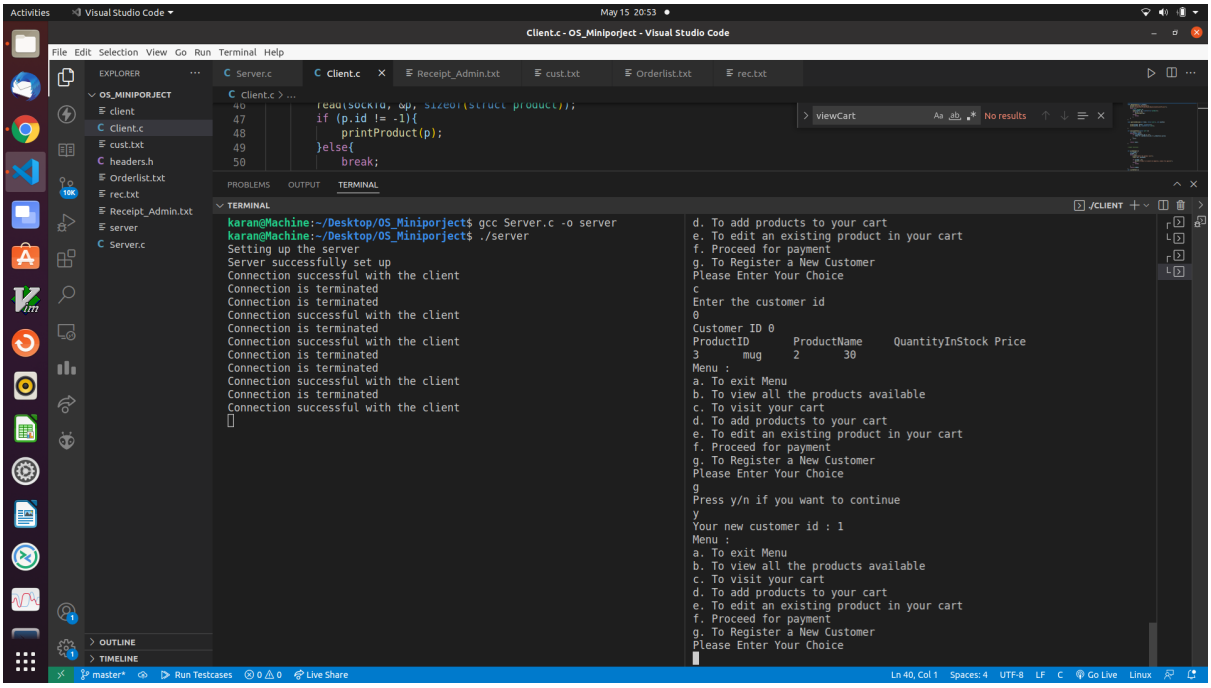
In a separate terminal, run the following commands
```
gcc -o client Client.c
./client
```

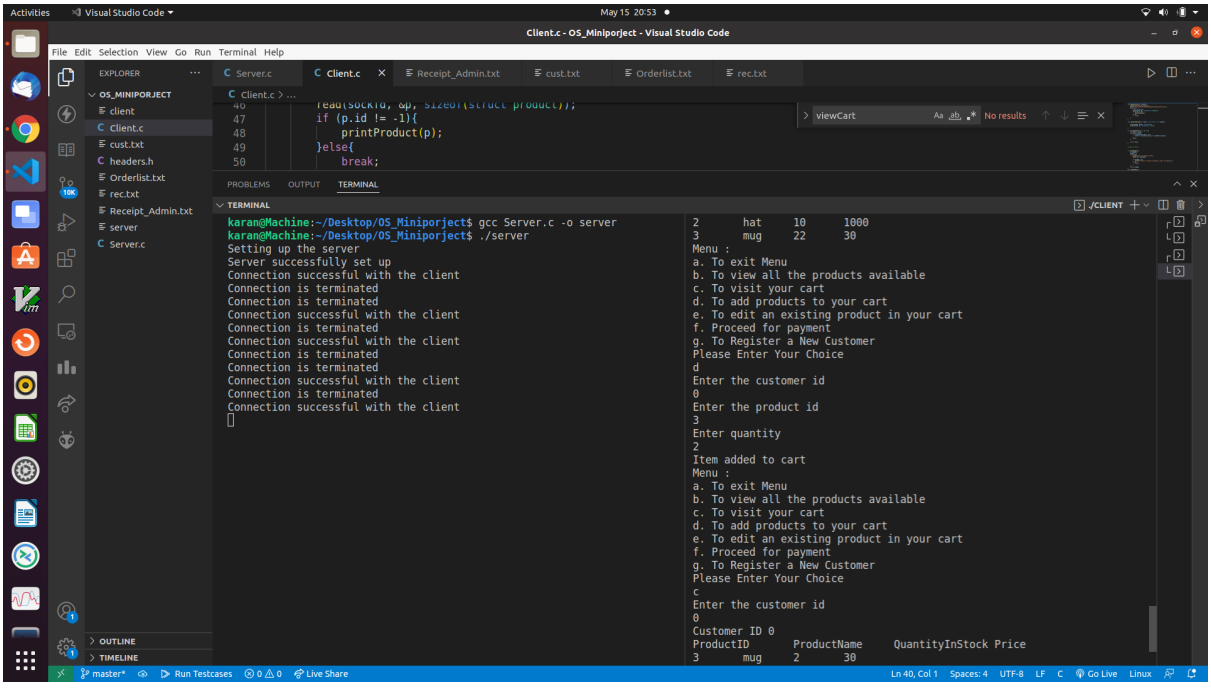
Now you can use the user menu or admin menu as directed by the program to perform operations on the products or customers.

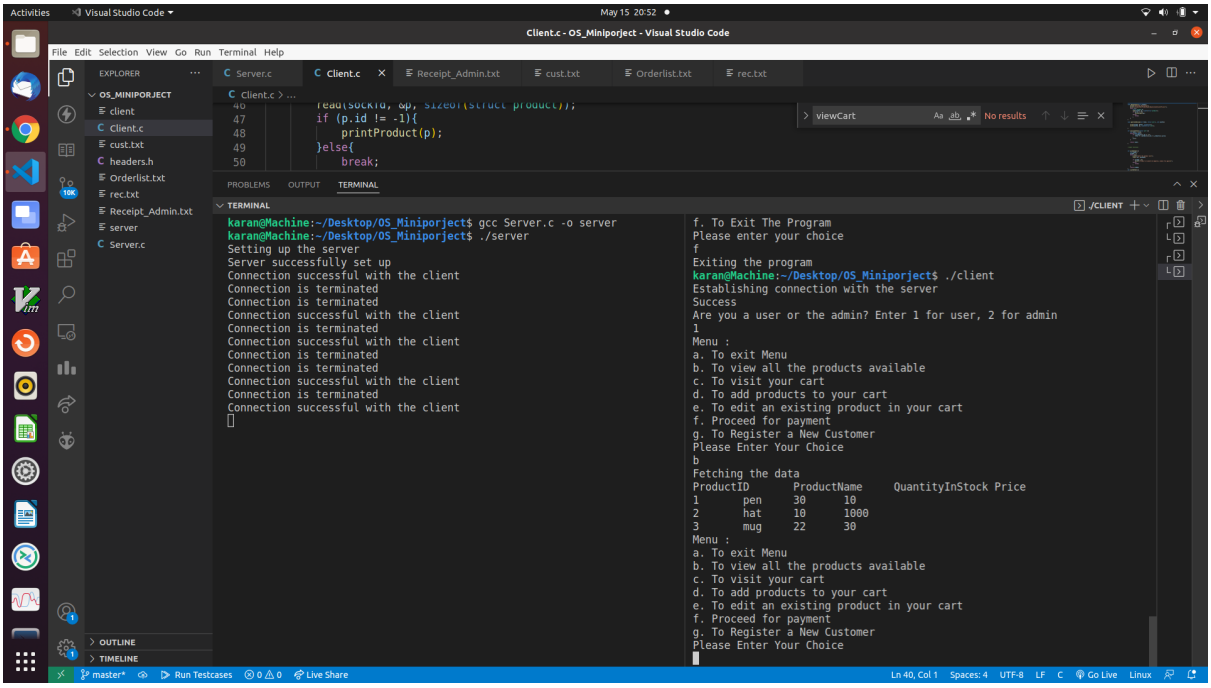
Screenshots of the results:

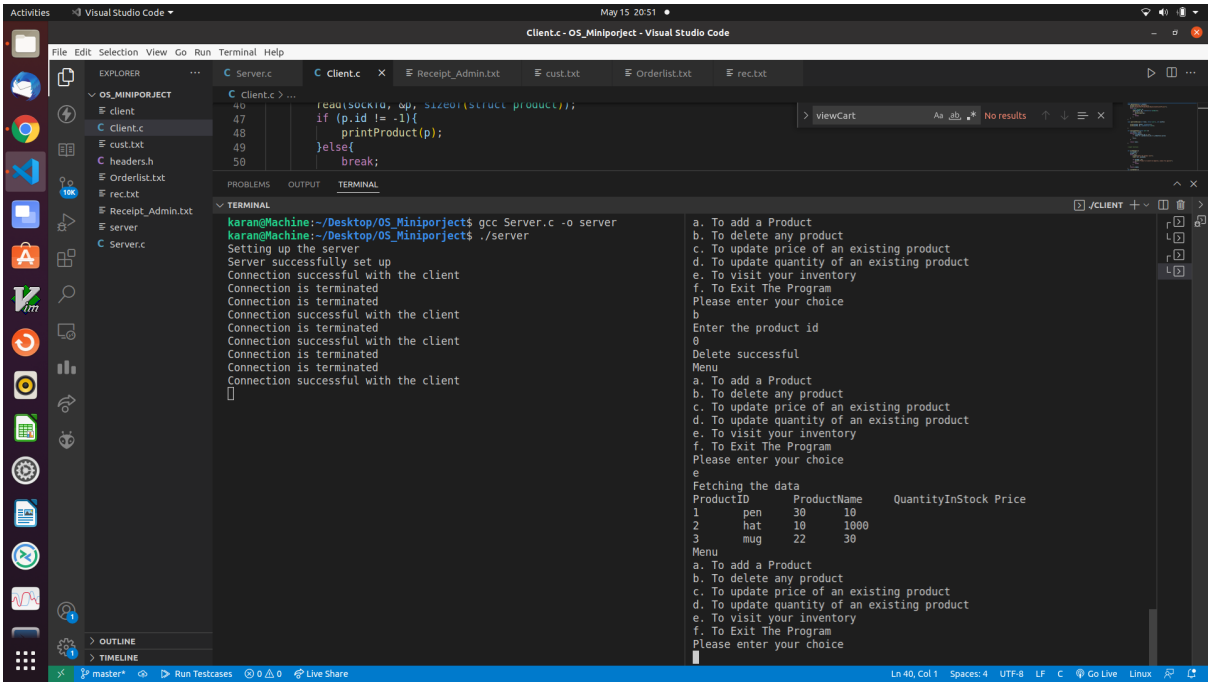


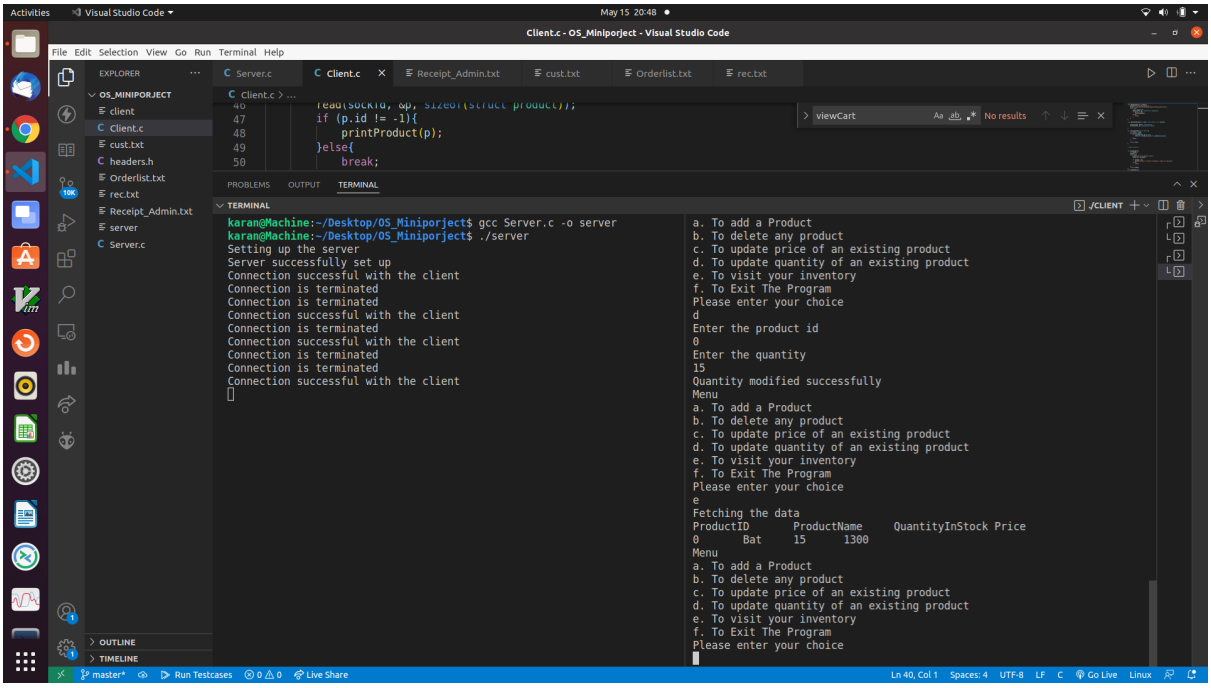


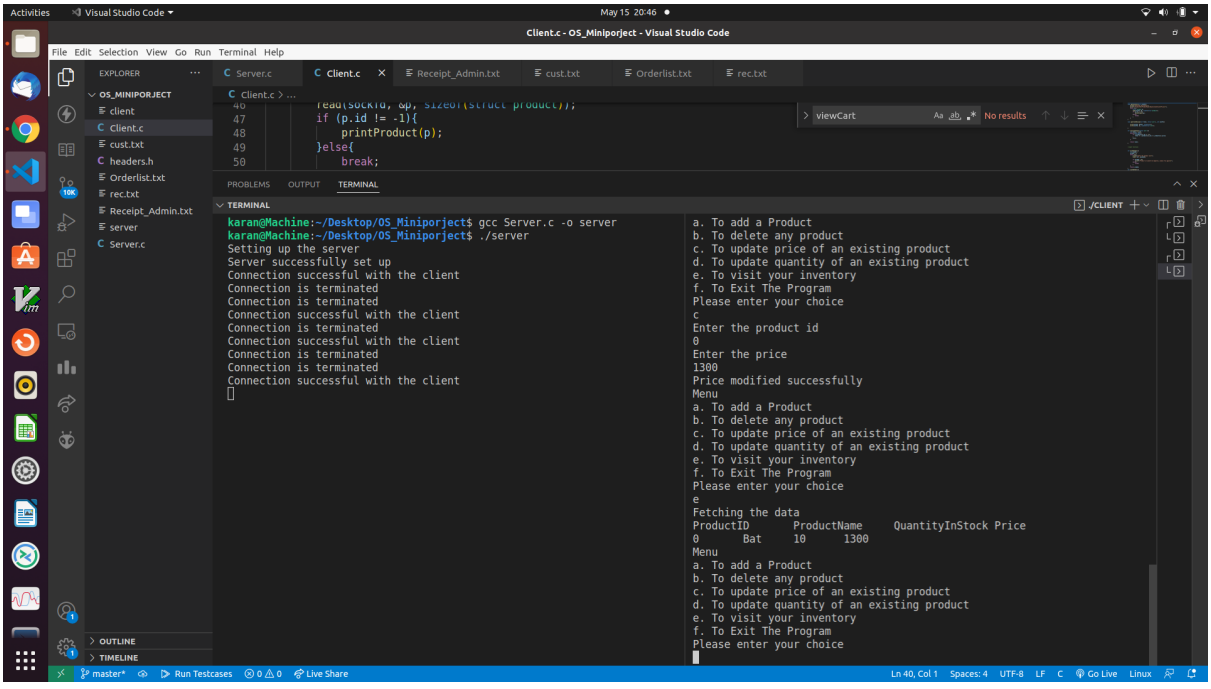

Activities
Visual Studio Code
May 15 20:53
Client.c - OS_Miniporject - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
OS_MINIPORJECT
client
Client.c
cust.txt
headers.h
Orderlist.txt
rec.txt
Receipt_Admin.txt
server
Server.c
Server.c
Client.c
Receipt_Admin.txt
cust.txt
Orderlist.txt
rec.txt
Client.c > ...
47 if (p.id != -1){
48 printProduct(p);
49 }else{
50 break;
viewCart
No results
PROBLEMS
OUTPUT
TERMINAL
TERMINAL
./client
karan@Machine:~/Desktop/OS_Miniporject$ gcc Server.c -o server
karan@Machine:~/Desktop/OS_Miniporject$ ./server
Setting up the server
Server successfully set up
Connection successful with the client
Connection is terminated
Connection is terminated
Connection successful with the client
Connection is terminated
Connection successful with the client
Connection is terminated
Connection successful with the client
Connection is terminated
Connection successful with the client
d. To add products to your cart
e. To edit an existing product in your cart
f. Proceed for payment
g. To Register a New Customer
Please Enter Your Choice
c
Enter the customer id
0
Customer ID 0
ProductID ProductName QuantityInStock Price
3 mug 2 30
Menu :
a. To exit Menu
b. To view all the products available
c. To visit your cart
d. To add products to your cart
e. To edit an existing product in your cart
f. Proceed for payment
g. To Register a New Customer
Please Enter Your Choice
g
Press y/n if you want to continue
y
Your new customer id : 1
Menu :
a. To exit Menu
b. To view all the products available
c. To visit your cart
d. To add products to your cart
e. To edit an existing product in your cart
f. Proceed for payment
g. To Register a New Customer
Please Enter Your Choice
OUTLINE
TIMELINE
master*
Run Testcases
0 0
Live Share
Ln 40, Col 1
Spaces: 4
UTF-8
LF
C
Go Live
Linux

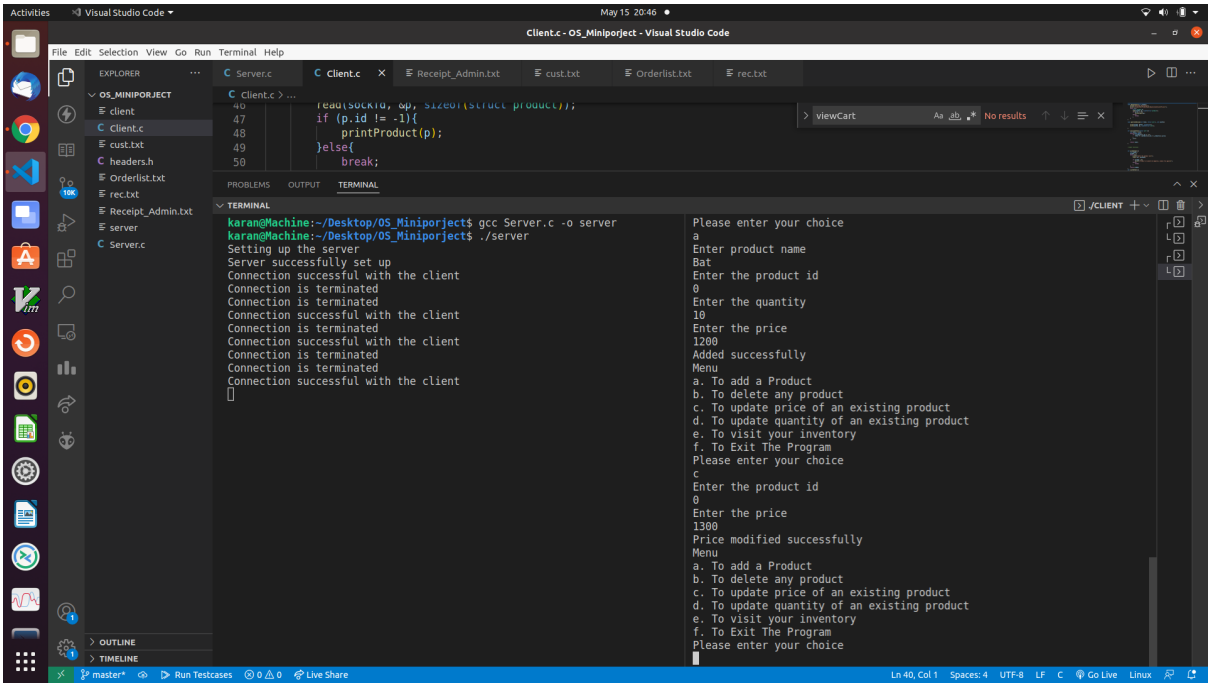

Activities
Visual Studio Code
May 15 20:53
Client.c - OS_Miniporject - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
OS_MINIPORJECT
client
Client.c
cust.txt
headers.h
Orderlist.txt
rec.txt
Receipt_Admin.txt
server
Server.c
Server.c
Client.c
Receipt_Admin.txt
cust.txt
Orderlist.txt
rec.txt
Client.c > ...
47 if (p.id != -1){
48 printProduct(p);
49 }else{
50 break;
viewCart
No results
PROBLEMS
OUTPUT
TERMINAL
TERMINAL
./client
karan@Machine:~/Desktop/OS_Miniporject$ gcc Server.c -o server
karan@Machine:~/Desktop/OS_Miniporject$ ./server
Setting up the server
Server successfully set up
Connection successful with the client
Connection is terminated
Connection is terminated
Connection successful with the client
Connection is terminated
Connection successful with the client
Connection is terminated
Connection successful with the client
Connection is terminated
Connection successful with the client
2 hat 10 1000
3 mug 22 30
Menu :
a. To exit Menu
b. To view all the products available
c. To visit your cart
d. To add products to your cart
e. To edit an existing product in your cart
f. Proceed for payment
g. To Register a New Customer
Please Enter Your Choice
d
Enter the customer id
0
Enter the product id
3
Enter quantity
2
Item added to cart
Menu :
a. To exit Menu
b. To view all the products available
c. To visit your cart
d. To add products to your cart
e. To edit an existing product in your cart
f. Proceed for payment
g. To Register a New Customer
Please Enter Your Choice
c
Enter the customer id
0
Customer ID 0
ProductID ProductName QuantityInStock Price
3 mug 2 30
OUTLINE
TIMELINE
master*
Run Testcases
0 0
Live Share
Ln 40, Col 1
Spaces: 4
UTF-8
LF
C
Go Live
Linux

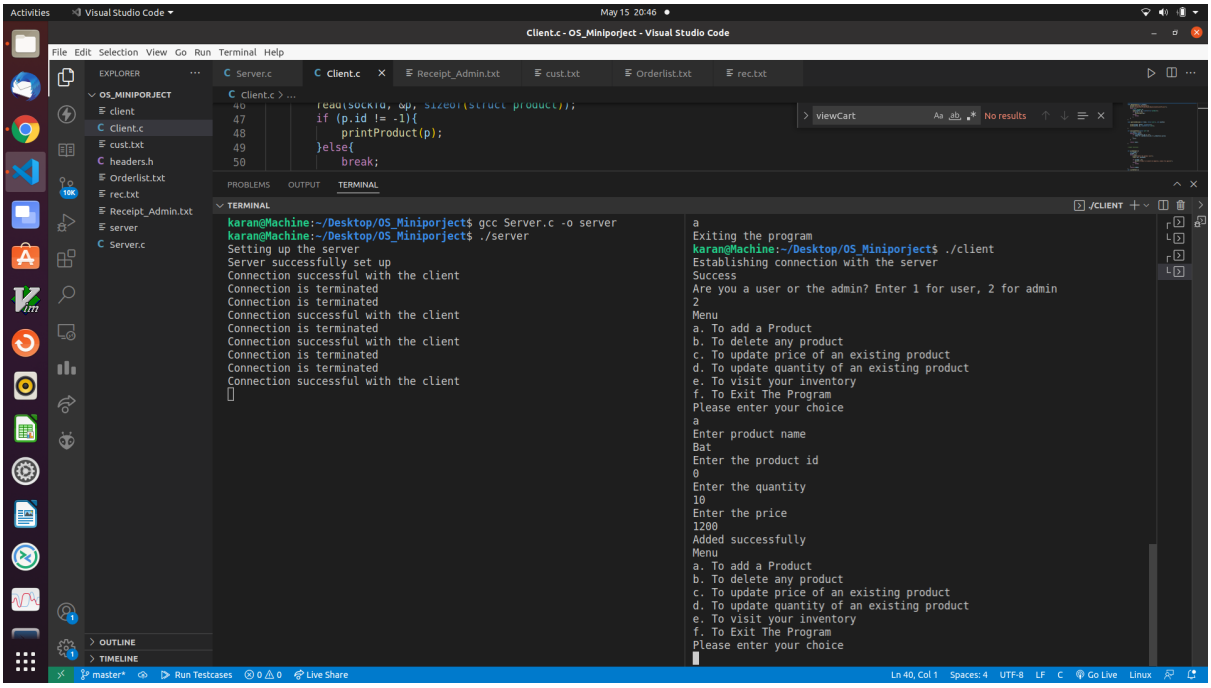

Activities
Visual Studio Code
May 15 20:52
Client.c - OS_Miniporject - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
OS_MINIPORJECT
client
Client.c
cust.txt
headers.h
Orderlist.txt
rec.txt
Receipt_Admin.txt
server
Server.c
Server.c
Client.c
Receipt_Admin.txt
cust.txt
Orderlist.txt
rec.txt
Client.c > ...
if (p.id != -1){
printProduct(p);
}else{
break;
viewCart
No results
PROBLEMS
OUTPUT
TERMINAL
TERMINAL
./CLIENT
karan@Machine:~/Desktop/OS_Miniporject$ gcc Server.c -o server
karan@Machine:~/Desktop/OS_Miniporject$ ./server
Setting up the server
Server successfully set up
Connection successful with the client
Connection is terminated
Connection is terminated
Connection successful with the client
Connection is terminated
Connection successful with the client
Connection is terminated
Connection successful with the client
Connection is terminated
Connection successful with the client
f. To Exit The Program
Please enter your choice
f
Exiting the program
karan@Machine:~/Desktop/OS_Miniporject$ ./client
Establishing connection with the server
Success
Are you a user or the admin? Enter 1 for user, 2 for admin
1
Menu :
a. To exit Menu
b. To view all the products available
c. To visit your cart
d. To add products to your cart
e. To edit an existing product in your cart
f. Proceed for payment
g. To Register a New Customer
Please Enter Your Choice
b
Fetching the data
ProductID ProductName QuantityInStock Price
1 pen 30 10
2 hat 10 1000
3 mug 22 30
Menu :
a. To exit Menu
b. To view all the products available
c. To visit your cart
d. To add products to your cart
e. To edit an existing product in your cart
f. Proceed for payment
g. To Register a New Customer
Please Enter Your Choice
OUTLINE
TIMELINE
master*
Run Testcases
Live Share
Ln 40, Col 1
Spaces: 4
UTF-8
LF
C
Go Live
Linux

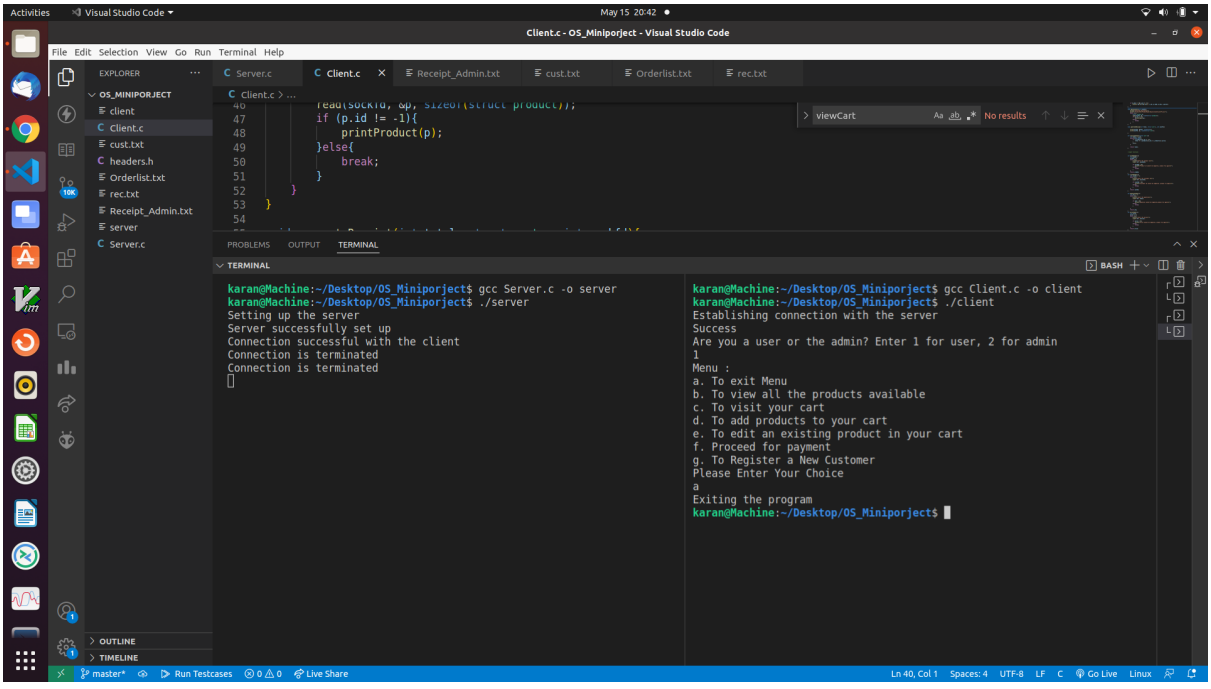

Activities
Visual Studio Code
May 15 20:51
Client.c - OS_Miniporject - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
OS_MINIPORJECT
client
Client.c
cust.txt
headers.h
Orderlist.txt
rec.txt
Receipt_Admin.txt
server
Server.c
Server.c
Client.c
Receipt_Admin.txt
cust.txt
Orderlist.txt
rec.txt
Client.c > ...
if (p.id != -1){
printProduct(p);
}else{
break;
viewCart
No results
PROBLEMS
OUTPUT
TERMINAL
TERMINAL
./CLIENT
karan@Machine:~/Desktop/OS_Miniporject$ gcc Server.c -o server
karan@Machine:~/Desktop/OS_Miniporject$ ./server
Setting up the server
Server successfully set up
Connection successful with the client
Connection is terminated
Connection is terminated
Connection successful with the client
Connection is terminated
Connection successful with the client
Connection is terminated
Connection successful with the client
Connection is terminated
Connection successful with the client
a. To add a Product
b. To delete any product
c. To update price of an existing product
d. To update quantity of an existing product
e. To visit your inventory
f. To Exit The Program
Please enter your choice
b
Enter the product id
0
Delete successful
Menu
a. To add a Product
b. To delete any product
c. To update price of an existing product
d. To update quantity of an existing product
e. To visit your inventory
f. To Exit The Program
Please enter your choice
e
Fetching the data
ProductID ProductName QuantityInStock Price
1 pen 30 10
2 hat 10 1000
3 mug 22 30
Menu
a. To add a Product
b. To delete any product
c. To update price of an existing product
d. To update quantity of an existing product
e. To visit your inventory
f. To Exit The Program
Please enter your choice
OUTLINE
TIMELINE
master*
Run Testcases
Live Share
Ln 40, Col 1
Spaces: 4
UTF-8
LF
C
Go Live
Linux


Activities
Visual Studio Code
May 15 20:48
Client.c - OS_Miniporject - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
OS_MINIPORJECT
client
Client.c
cust.txt
headers.h
Orderlist.txt
rec.txt
Receipt_Admin.txt
server
Server.c
Server.c
Client.c
Receipt_Admin.txt
cust.txt
Orderlist.txt
rec.txt
Client.c > ...
47 if (p.id != -1){
48 printProduct(p);
49 }else{
50 break;
viewCart
No results
PROBLEMS
OUTPUT
TERMINAL
TERMINAL
./client
karan@Machine:~/Desktop/OS_Miniporject$ gcc Server.c -o server
karan@Machine:~/Desktop/OS_Miniporject$ ./server
Setting up the server
Server successfully set up
Connection successful with the client
Connection is terminated
Connection is terminated
Connection successful with the client
Connection is terminated
Connection successful with the client
Connection is terminated
Connection is terminated
Connection successful with the client
a. To add a Product
b. To delete any product
c. To update price of an existing product
d. To update quantity of an existing product
e. To visit your inventory
f. To Exit The Program
Please enter your choice
d
Enter the product id
0
Enter the quantity
15
Quantity modified successfully
Menu
a. To add a Product
b. To delete any product
c. To update price of an existing product
d. To update quantity of an existing product
e. To visit your inventory
f. To Exit The Program
Please enter your choice
e
Fetching the data
ProductID ProductName QuantityInStock Price
0 Bat 15 1300
Menu
a. To add a Product
b. To delete any product
c. To update price of an existing product
d. To update quantity of an existing product
e. To visit your inventory
f. To Exit The Program
Please enter your choice
OUTLINE
TIMELINE
master*
Run Testcases
Live Share
Ln 40, Col 1
Spaces: 4
UTF-8
LF
C
Go Live
Linux


Activities
Visual Studio Code
May 15 20:46
Client.c - OS_Miniporject - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
OS_MINIPORJECT
client
Client.c
cust.txt
headers.h
Orderlist.txt
rec.txt
Receipt_Admin.txt
server
Server.c
Server.c
Client.c
Receipt_Admin.txt
cust.txt
Orderlist.txt
rec.txt
Client.c > ...
47 if (p.id != -1){
48 printProduct(p);
49 }else{
50 break;
viewCart
No results
PROBLEMS
OUTPUT
TERMINAL
TERMINAL
./client
karan@Machine:~/Desktop/OS_Miniporject$ gcc Server.c -o server
karan@Machine:~/Desktop/OS_Miniporject$ ./server
Setting up the server
Server successfully set up
Connection successful with the client
Connection is terminated
Connection is terminated
Connection successful with the client
Connection is terminated
Connection successful with the client
Connection is terminated
Connection is terminated
Connection successful with the client
a. To add a Product
b. To delete any product
c. To update price of an existing product
d. To update quantity of an existing product
e. To visit your inventory
f. To Exit The Program
Please enter your choice
c
Enter the product id
0
Enter the price
1300
Price modified successfully
Menu
a. To add a Product
b. To delete any product
c. To update price of an existing product
d. To update quantity of an existing product
e. To visit your inventory
f. To Exit The Program
Please enter your choice
e
Fetching the data
ProductID ProductName QuantityInStock Price
0 Bat 10 1300
Menu
a. To add a Product
b. To delete any product
c. To update price of an existing product
d. To update quantity of an existing product
e. To visit your inventory
f. To Exit The Program
Please enter your choice
OUTLINE
TIMELINE
master*
Run Testcases
Live Share
Ln 40, Col 1
Spaces: 4
UTF-8
LF
C
Go Live
Linux


Activities
Visual Studio Code
May 15 20:46
Client.c - OS_Miniporject - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
OS_MINIPORJECT
client
Client.c
cust.txt
headers.h
Orderlist.txt
rec.txt
Receipt_Admin.txt
server
Server.c
Server.c
Client.c
Receipt_Admin.txt
cust.txt
Orderlist.txt
rec.txt
Client.c > ...
if (p.id != -1){
printProduct(p);
}else{
break;
viewCart
No results
PROBLEMS
OUTPUT
TERMINAL
TERMINAL
./CLIENT
karan@Machine:~/Desktop/OS_Miniporject$ gcc Server.c -o server
karan@Machine:~/Desktop/OS_Miniporject$ ./server
Setting up the server
Server successfully set up
Connection successful with the client
Connection is terminated
Connection is terminated
Connection successful with the client
Connection is terminated
Connection successful with the client
Connection is terminated
Connection is terminated
Connection successful with the client
Please enter your choice
a
Enter product name
Bat
Enter the product id
0
Enter the quantity
10
Enter the price
1200
Added successfully
Menu
a. To add a Product
b. To delete any product
c. To update price of an existing product
d. To update quantity of an existing product
e. To visit your inventory
f. To Exit The Program
Please enter your choice
c
Enter the product id
0
Enter the price
1300
Price modified successfully
Menu
a. To add a Product
b. To delete any product
c. To update price of an existing product
d. To update quantity of an existing product
e. To visit your inventory
f. To Exit The Program
Please enter your choice
OUTLINE
TIMELINE
master*
Run Testcases
0 0
Live Share
Ln 40, Col 1
Spaces: 4
UTF-8
LF
C
Go Live
Linux


Activities
Visual Studio Code
May 15 20:46
Client.c - OS_Miniporject - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
OS_MINIPORJECT
client
Client.c
cust.txt
headers.h
Orderlist.txt
rec.txt
Receipt_Admin.txt
server
Server.c
Server.c
Client.c
Receipt_Admin.txt
cust.txt
Orderlist.txt
rec.txt
viewCart
No results
47 if (p.id != -1){
48 printProduct(p);
49 }else{
50 break;
PROBLEMS
OUTPUT
TERMINAL
karan@Machine:~/Desktop/OS_Miniporject$ gcc Server.c -o server
karan@Machine:~/Desktop/OS_Miniporject$ ./server
Setting up the server
Server successfully set up
Connection successful with the client
Connection is terminated
Connection is terminated
Connection successful with the client
Connection is terminated
Connection successful with the client
Connection is terminated
Connection is terminated
Connection successful with the client
a
Exiting the program
karan@Machine:~/Desktop/OS_Miniporject$ ./client
Establishing connection with the server
Success
Are you a user or the admin? Enter 1 for user, 2 for admin
2
Menu
a. To add a Product
b. To delete any product
c. To update price of an existing product
d. To update quantity of an existing product
e. To visit your inventory
f. To Exit The Program
Please enter your choice
a
Enter product name
Bat
Enter the product id
0
Enter the quantity
10
Enter the price
1200
Added successfully
Menu
a. To add a Product
b. To delete any product
c. To update price of an existing product
d. To update quantity of an existing product
e. To visit your inventory
f. To Exit The Program
Please enter your choice
OUTLINE
TIMELINE
master*
Run Testcases
Live Share
Ln 40, Col 1
Spaces: 4
UTF-8
LF
C
Go Live
Linux


Activities
Visual Studio Code
May 15 20:42
Client.c - OS_Miniporject - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
OS_MINIPORJECT
client
Client.c
cust.txt
headers.h
Orderlist.txt
rec.txt
Receipt_Admin.txt
server
Server.c
Server.c
Client.c
Receipt_Admin.txt
cust.txt
Orderlist.txt
rec.txt
viewCart
No results
47 if (p.id != -1){
48 printProduct(p);
49 }else{
50 break;
51 }
52 }
53 }
PROBLEMS
OUTPUT
TERMINAL
karan@Machine:~/Desktop/OS_Miniporject$ gcc Server.c -o server
karan@Machine:~/Desktop/OS_Miniporject$ ./server
Setting up the server
Server successfully set up
Connection successful with the client
Connection is terminated
Connection is terminated
karan@Machine:~/Desktop/OS_Miniporject$ gcc Client.c -o client
karan@Machine:~/Desktop/OS_Miniporject$ ./client
Establishing connection with the server
Success
Are you a user or the admin? Enter 1 for user, 2 for admin
1
Menu :
a. To exit Menu
b. To view all the products available
c. To visit your cart
d. To add products to your cart
e. To edit an existing product in your cart
f. Proceed for payment
g. To Register a New Customer
Please Enter Your Choice
a
Exiting the program
karan@Machine:~/Desktop/OS_Miniporject$
OUTLINE
TIMELINE
master*
Run Testcases
Live Share
Ln 40, Col 1
Spaces: 4
UTF-8
LF
C
Go Live
Linux