OS Mini project – 2023

Name: Achintya Harsha

Roll No: IMT2021525

Email: achintya.harsha@iiitb.ac.in

GitHub link: https://github.com/achintyapro03/OS-Project

Overview:

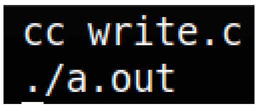
This project is a terminal oriented project which implements client – server based online store application. The user / admin can login from the client portal and the server is responsible for handling the requests sent by the client. The connection between the client and the sever is implemented using socket programming. The server is also a concurrent server i.e., it can handle multiple clients simultaneously. This was implemented by creating a new fork for every new client. Finally, the concept of file locking has been implemented so that multiple users cannot change or access the same file or record in the table as otherwise it might lead to date inconsistency.

Project/File Architecture:

1. **Server.c:** Implements the server-side code that accepts requests from the client processes the requests by calling the appropriate functions and returns the response back to the client. The request and response information are communicated using socket programming.
2. **Client.c:** Implements the client-side code by displaying the login menu to the user. Once logged in, a menu will be displayed to the user based on whether the user is an admin or a normal user. Based on the option selected by the user, a request is formulated and is sent to the server through a socket established by socket programming.
3. **DB Folder:** Db folder contains 3 files namely – userTable, productTable and cartTable. The files store the details of the user, product, and the cart information. The information is accessed and modified by the read and write system calls.
4. **Write.c:** For this project we are hard coding the admin user information. Write.c program writes the admin information to the userTable.
5. **data.h:** This file defines all the necessary structs (product, user, and cart) required for executing the program. (This file should not be compiled)

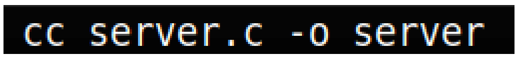
Steps to run:

1. Compile and run the write.c program to configure the admin details.

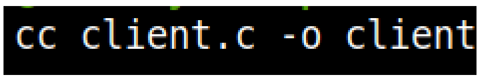


```
cc write.c
./a.out
```

2. Now compile the server.c file and execute the executable file. Similarly run the client.c file.



```
cc server.c -o server
```



```
cc client.c -o client
```

NOTE: Make sure to run the server.c file before the client.c file.

User Functionalities:

Based on the type of the user, different rights have been granted to different users accordingly.

Common functionalities:

1. **Login:** A user can login using his registered username and password. The server authenticates the user and if valid, the user is logged in. Else an error message is generated on the client side notifying the user of the failure.
2. **Register:** A user can register by entering his username and password. If a user with the same name already exists, then error message is generated. Else the user is logged in.
3. **View Products:** Admin/ User can select this option to get a detailed catalogue of the products available in the shop. This catalogue provides information about the productid, product name, quantity available and price.
4. **View one specific Product:** If information about a specific product is required, then this function can be used to get info about it. It takes product ID as an input.
5. **Logout:** Selecting this option logs out the user.

Admin Functionalities:

1. **Add product:** This feature enables the admin to add a product into the inventory.
2. **Delete product:** Allows the user to delete an item from the shop.
3. **Update product:** Allows the admin to update a specific product in the inventory (product price and quantity) based on the product id entered.

User Functionalities:

1. **Add to cart:** An item can be added to the cart based on the product id entered.

   An item can be added to the cart only if enough is available in the inventory.

2. **View Cart:** All the items added to the cart can be viewed using this feature.
3. **Edit Cart:** The quantity of the items added to the cart can be modified using this feature.
4. **Buy:** The user is also allowed to buy items he has added to the cart. It asks for a confirmation form user before buying. If confirmed, then the bill is displayed, and the user is expected to enter the amount for transaction. If the amount matches with the bill total, then the transaction is a success.

Flow of control in program:

Client

1. The main function initially establishes a socket connection with the user. Then it starts a while loop to handle the requests from the user.
2. Then it displays a login menu. Once logged in, based on whether, the user is admin or a normal user, 2 distinct types of menus is displayed to cater to unique needs.
3. The user can select the operation by entering the number associated with it. Also, some functions ask for extra details like the product id, name, quantity, and price.
4. Now a request is sent to the server in chunks. First the userId and the opcode of the function is sent. (Opcode – operation code is a unique code associated with the function and this enables to server to identify the type of request).
5. Then the other extra parameters are also inputted from the user and sent to the server via the socket.
6. When the server generates the response, the read system call reads the response and then prints it.
7. This is repeated in a loop until the user exits.

Server

1. The main function creates a socket using the socket system call. A while loop is started, so that new clients can be connected as and when a connection request is sent by the user. This makes use of the concept of forks to achieve a concurrent server.
2. When a client tries to connect with the server, a new process is created using fork and the linker () function is called.

3. This linker function has I while loop that keeps on reading requests from the user passes the request to a handler function (processRequest()) and returns an appropriate response.
4. The processRequest function first receives the userid and the opcode of the requests.
5. The userId for login or register is set as -1 as the user is still not logged in. Once the user has logged in the userId is sent back to the user for future use.
6. Next based on the opcode the relevant handler function is called. Now this handler function might need some other details as well. So again, using the read system call, the data is read from the socket.
7. Now these handler functions, after a series of instructions, generate a response which is written back to the socket and this entire process repeats repeatedly in a loop to handle new requests from the user.

Implementation of OS concepts:

1. **Socket programming**:
   The application uses socket programming to communicate between the user and client.
   a. The program uses server-side system calls (socket, bind, listen, accept) to setup the connection at the server side. In contrast, the client side uses – socket and the connect system calls to establish the connection.
   b. Fork is used to setup a concurrent server.
   c. All reads and writes from the socket uses "read" and "write" system calls.

2. **File Locking**:

   File locking is implemented so that the data in the database is concurrent. Two types of record locking have been implemented – Mandatory locking and Record locking.

   a. **Mandatory Locking** has been implemented in the following cases:
      i. When a new product is being added, the entire file is locked using the fcntl function. The lock configuration is set to Write lock and using the SETLKW mode.
      ii. When "View all product" is executed
   b. **Record Locking** has been implemented in the following cases:

      Product id has been implemented in such a way that it is sequential i.e., with addition of every new product, the product id is incremented by one.

      i. When "view 1 product" is selected, product id is taken as input, and that region is locked in the "productTable" file.

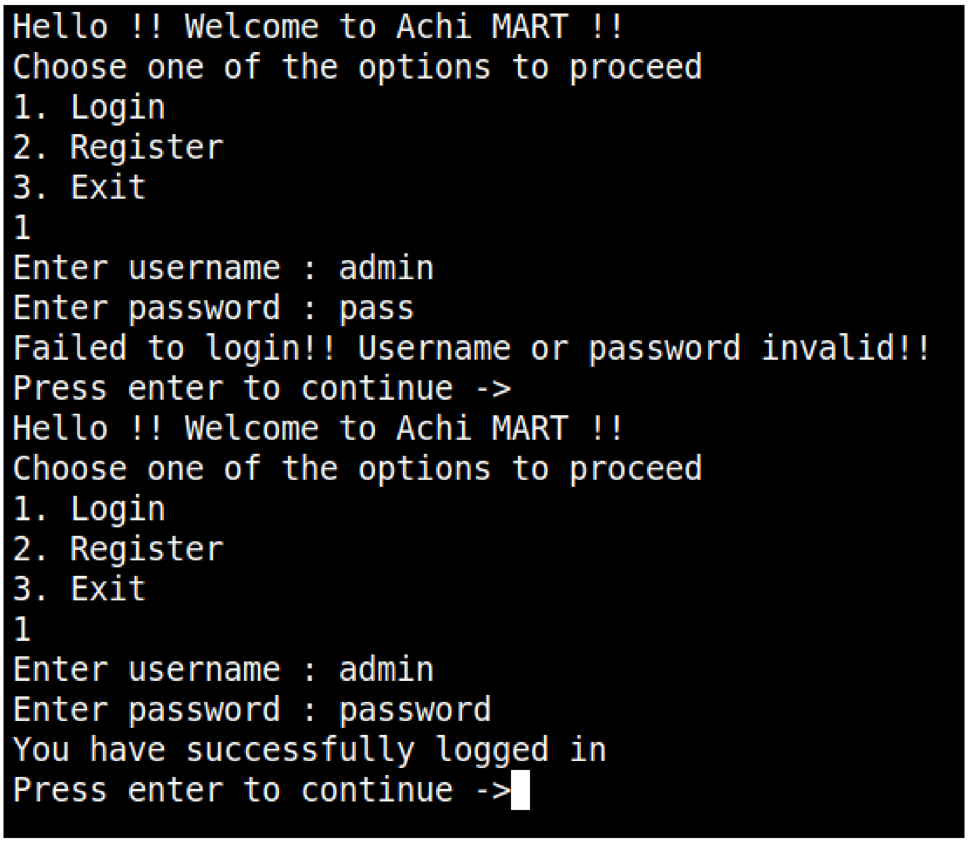
ii. When “delete product” is selected
iii. When “update product” is selected the record lock is implemented. Then further details like quantity and price are asked from user which is sent through the socket. This is read by the read system call. The server stalls until the client enters the necessary details. Once the details are entered, the updateProduct function is called and the lock is released after the execution of the function.
iv. When “View Cart” is executed
v. When “Add to cart” is executed
vi. When “buy” is executed

3. **File handing**:

System calls like open, read, and write were used in the program to implement file handling.

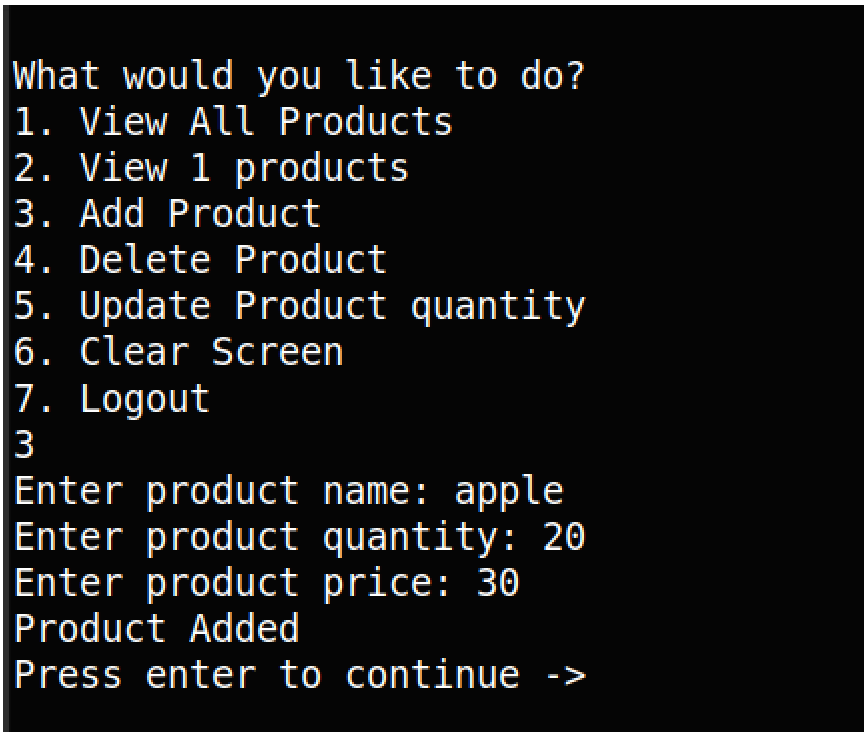
Screenshots:

1. Login (admin login)



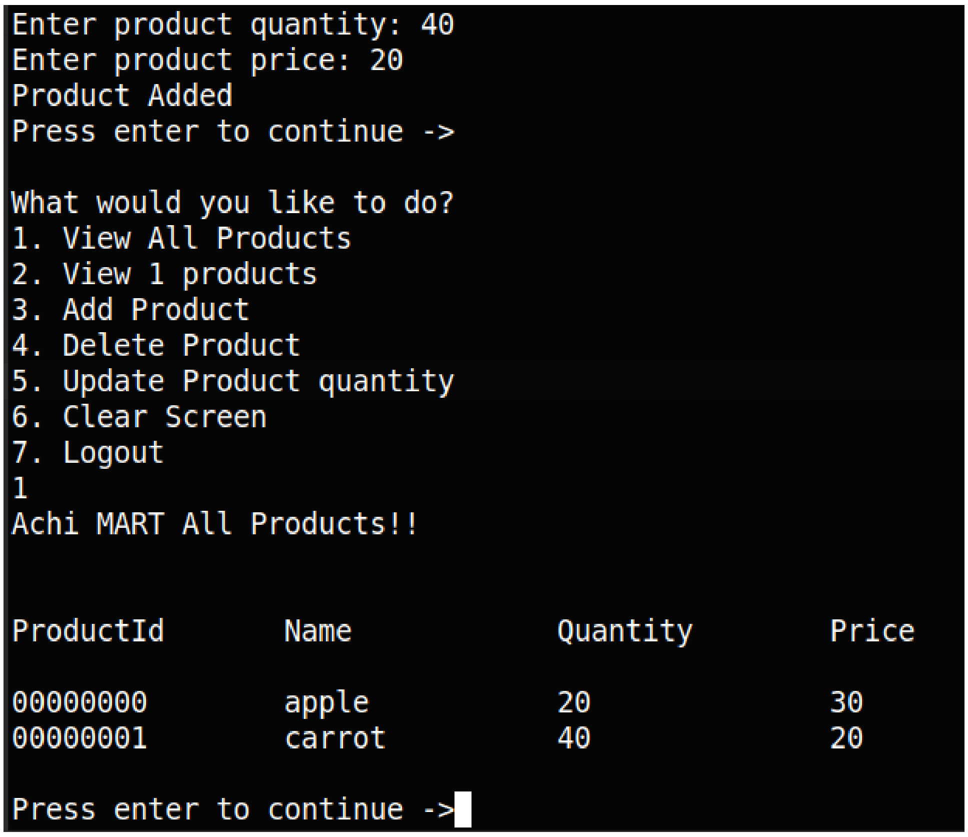
```
Hello !! Welcome to Achi MART !!
Choose one of the options to proceed
1. Login
2. Register
3. Exit
1
Enter username : admin
Enter password : pass
Failed to login!! Username or password invalid!!
Press enter to continue ->
Hello !! Welcome to Achi MART !!
Choose one of the options to proceed
1. Login
2. Register
3. Exit
1
Enter username : admin
Enter password : password
You have successfully logged in
Press enter to continue ->
```

2. Add product



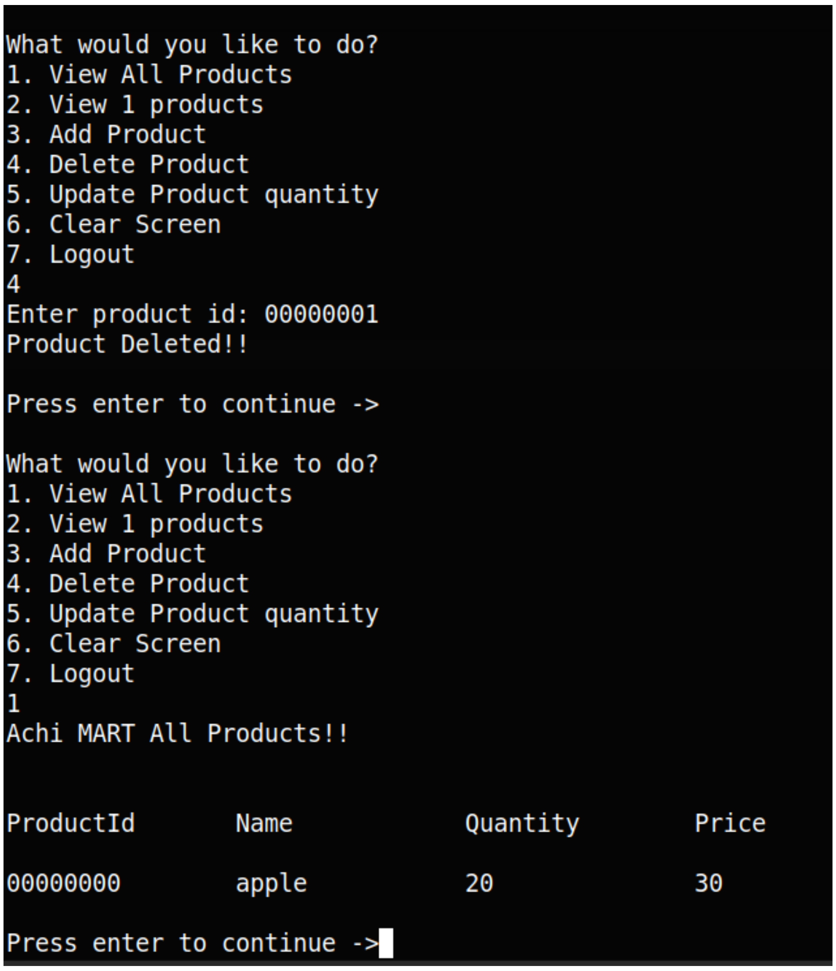
```
What would you like to do?
1. View All Products
2. View 1 products
3. Add Product
4. Delete Product
5. Update Product quantity
6. Clear Screen
7. Logout
3
Enter product name: apple
Enter product quantity: 20
Enter product price: 30
Product Added
Press enter to continue ->
```

3. View All Products



```
Enter product quantity: 40
Enter product price: 20
Product Added
Press enter to continue ->

What would you like to do?
1. View All Products
2. View 1 products
3. Add Product
4. Delete Product
5. Update Product quantity
6. Clear Screen
7. Logout
1
Achi MART All Products!!


ProductId       Name            Quantity        Price

00000000        apple           20              30
00000001        carrot          40              20

Press enter to continue ->
```

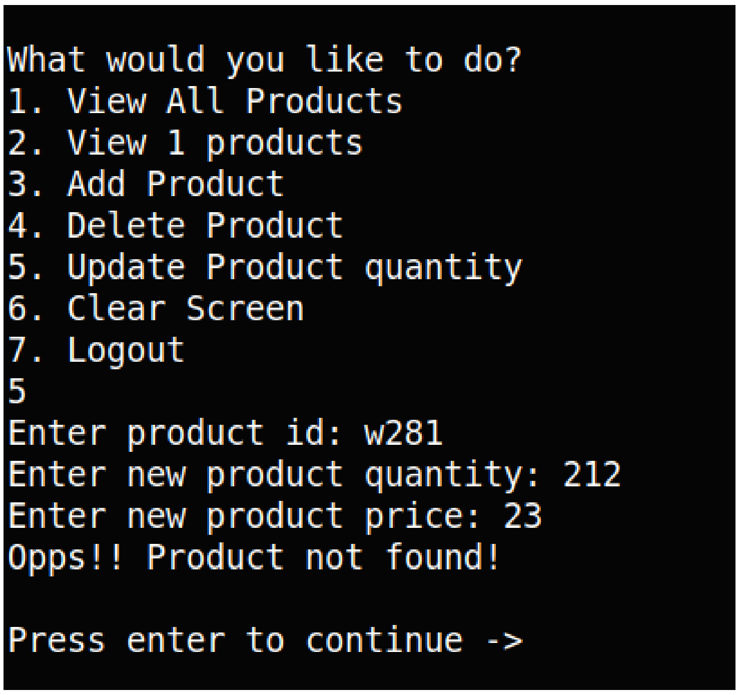
4. Delete Product



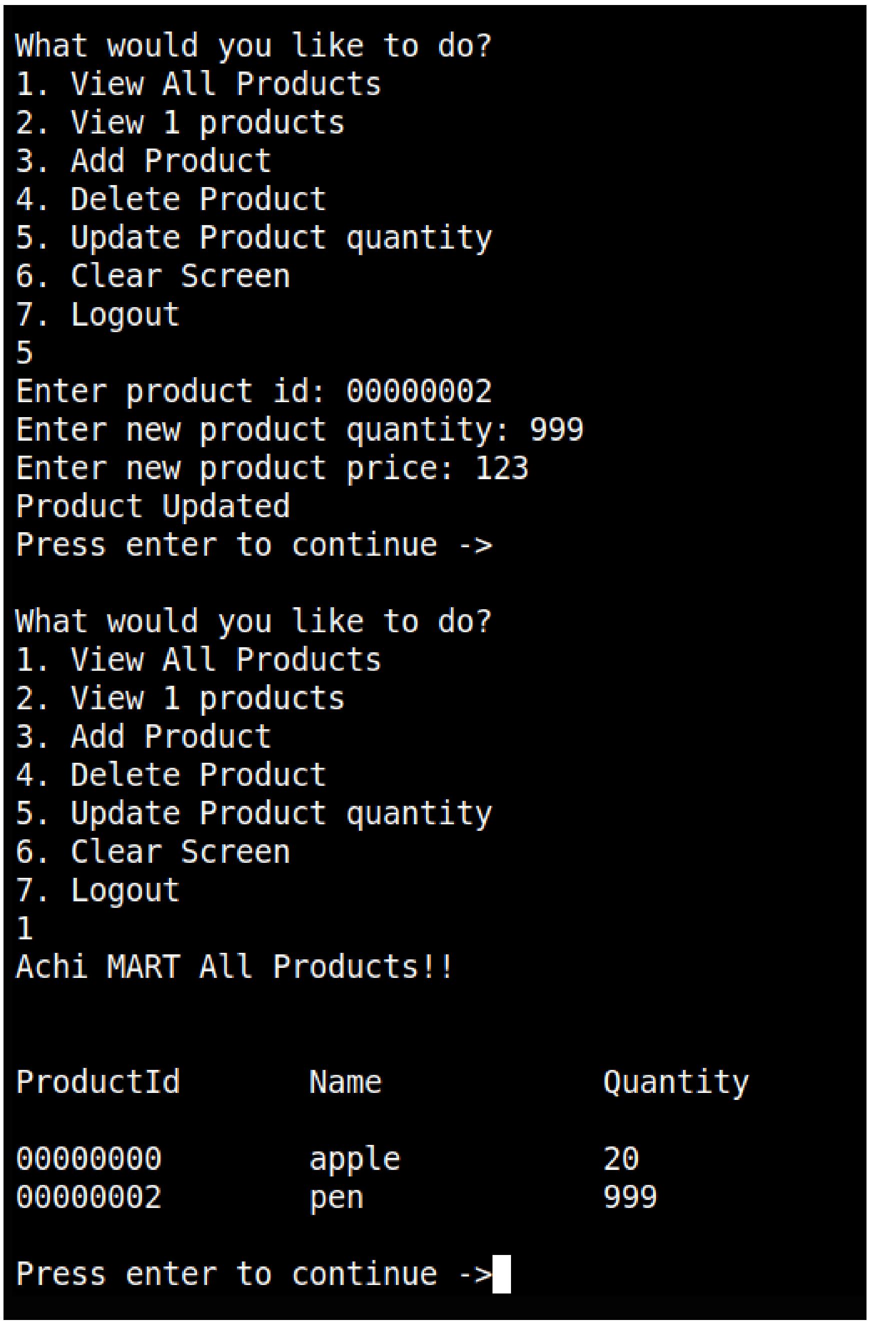
```
What would you like to do?
1. View All Products
2. View 1 products
3. Add Product
4. Delete Product
5. Update Product quantity
6. Clear Screen
7. Logout
4
Enter product id: 00000001
Product Deleted!!

Press enter to continue ->

What would you like to do?
1. View All Products
2. View 1 products
3. Add Product
4. Delete Product
5. Update Product quantity
6. Clear Screen
7. Logout
1
Achi MART All Products!!


ProductId       Name            Quantity        Price

00000000        apple           20              30

Press enter to continue ->
```

5. Update Product

   Invalid product id is entered. Hence an error message is displayed.

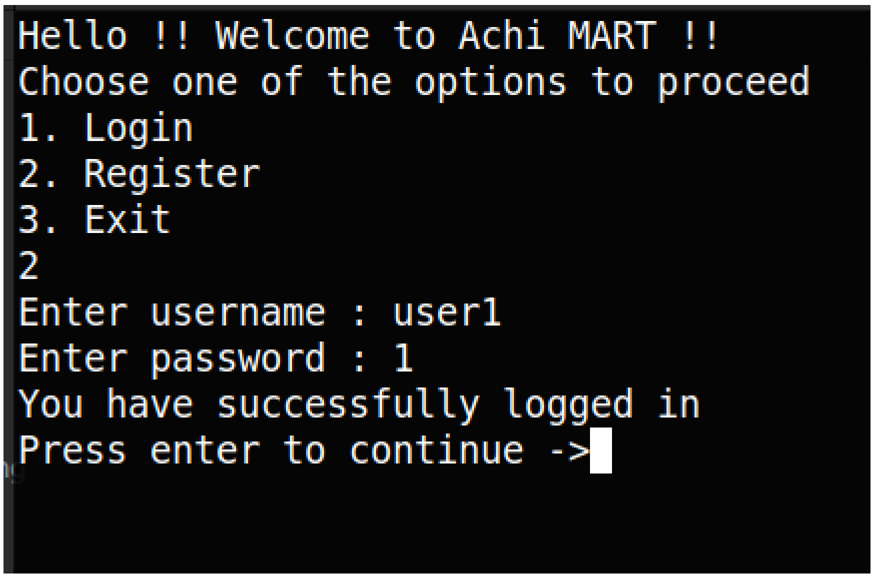


```
What would you like to do?
1. View All Products
2. View 1 products
3. Add Product
4. Delete Product
5. Update Product quantity
6. Clear Screen
7. Logout
5
Enter product id: w281
Enter new product quantity: 212
Enter new product price: 23
Opps!! Product not found!

Press enter to continue ->
```



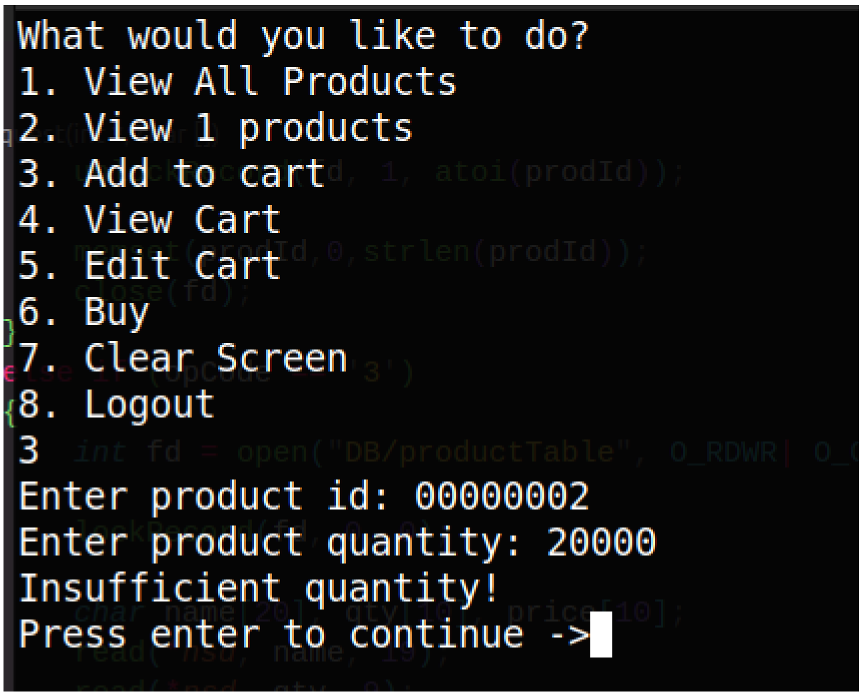
```
What would you like to do?
1. View All Products
2. View 1 products
3. Add Product
4. Delete Product
5. Update Product quantity
6. Clear Screen
7. Logout
5
Enter product id: 00000002
Enter new product quantity: 999
Enter new product price: 123
Product Updated
Press enter to continue ->

What would you like to do?
1. View All Products
2. View 1 products
3. Add Product
4. Delete Product
5. Update Product quantity
6. Clear Screen
7. Logout
1
Achi MART All Products!!


ProductId       Name            Quantity

00000000        apple           20
00000002        pen             999

Press enter to continue ->
```

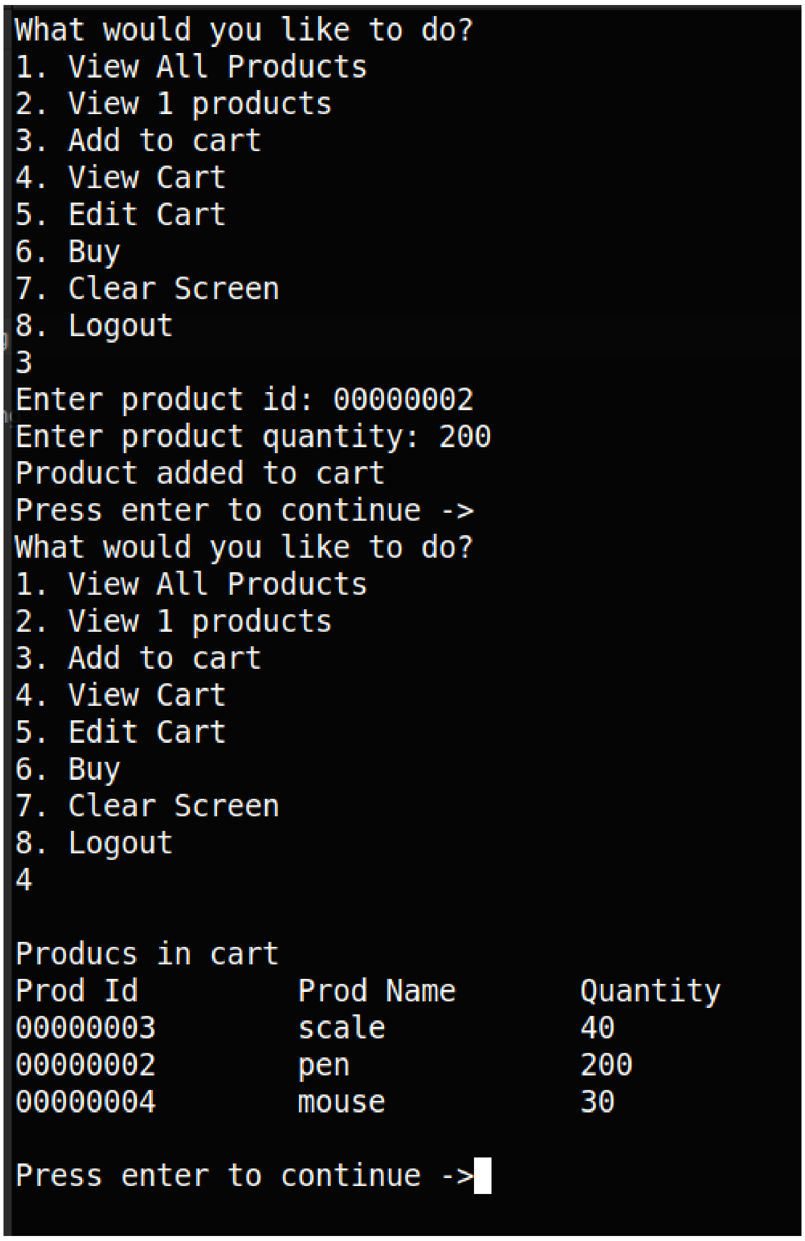
6. Register (User login)


```
Hello !! Welcome to Achi MART !!
Choose one of the options to proceed
1. Login
2. Register
3. Exit
2
Enter username : user1
Enter password : 1
You have successfully logged in
Press enter to continue ->
```

7. Add to cart

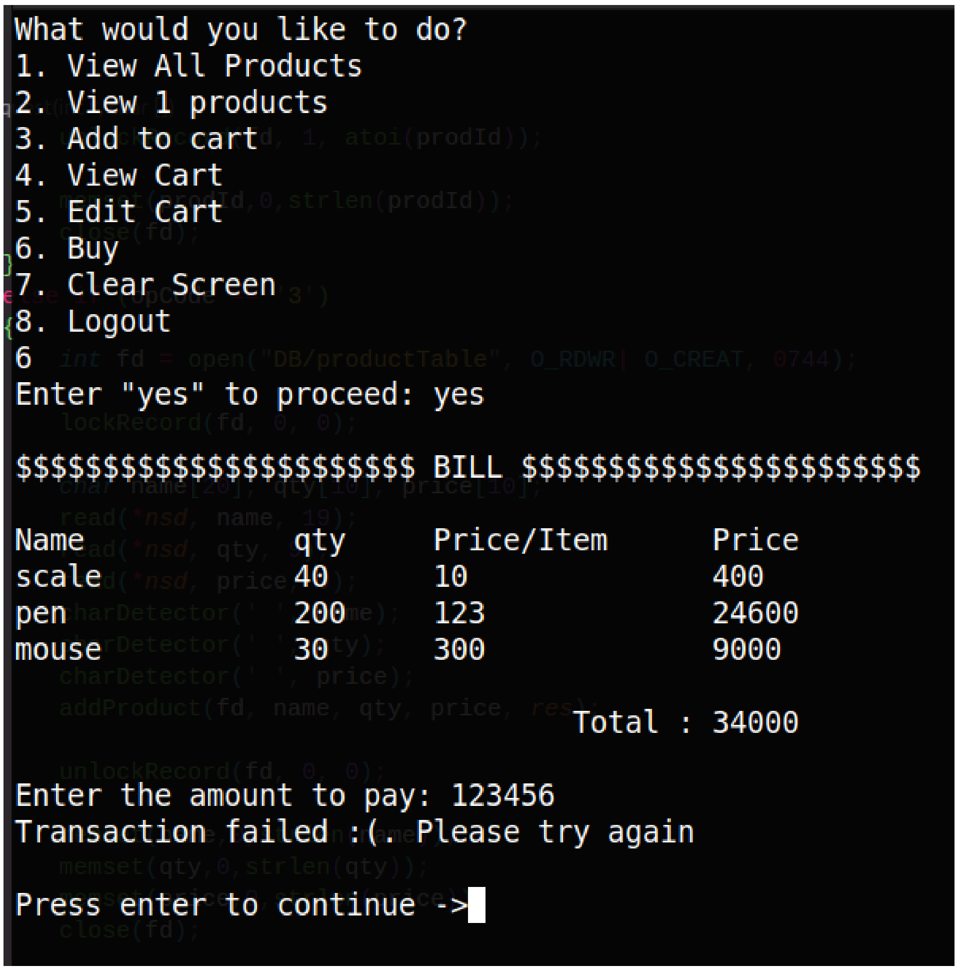
The quantity entered was not available in the inventory. Hence insufficient quantity is displayed.


```
What would you like to do?
1. View All Products
2. View 1 products
3. Add to cart
4. View Cart
5. Edit Cart
6. Buy
7. Clear Screen
8. Logout
3
Enter product id: 00000002
Enter product quantity: 20000
Insufficient quantity!
Press enter to continue ->
```



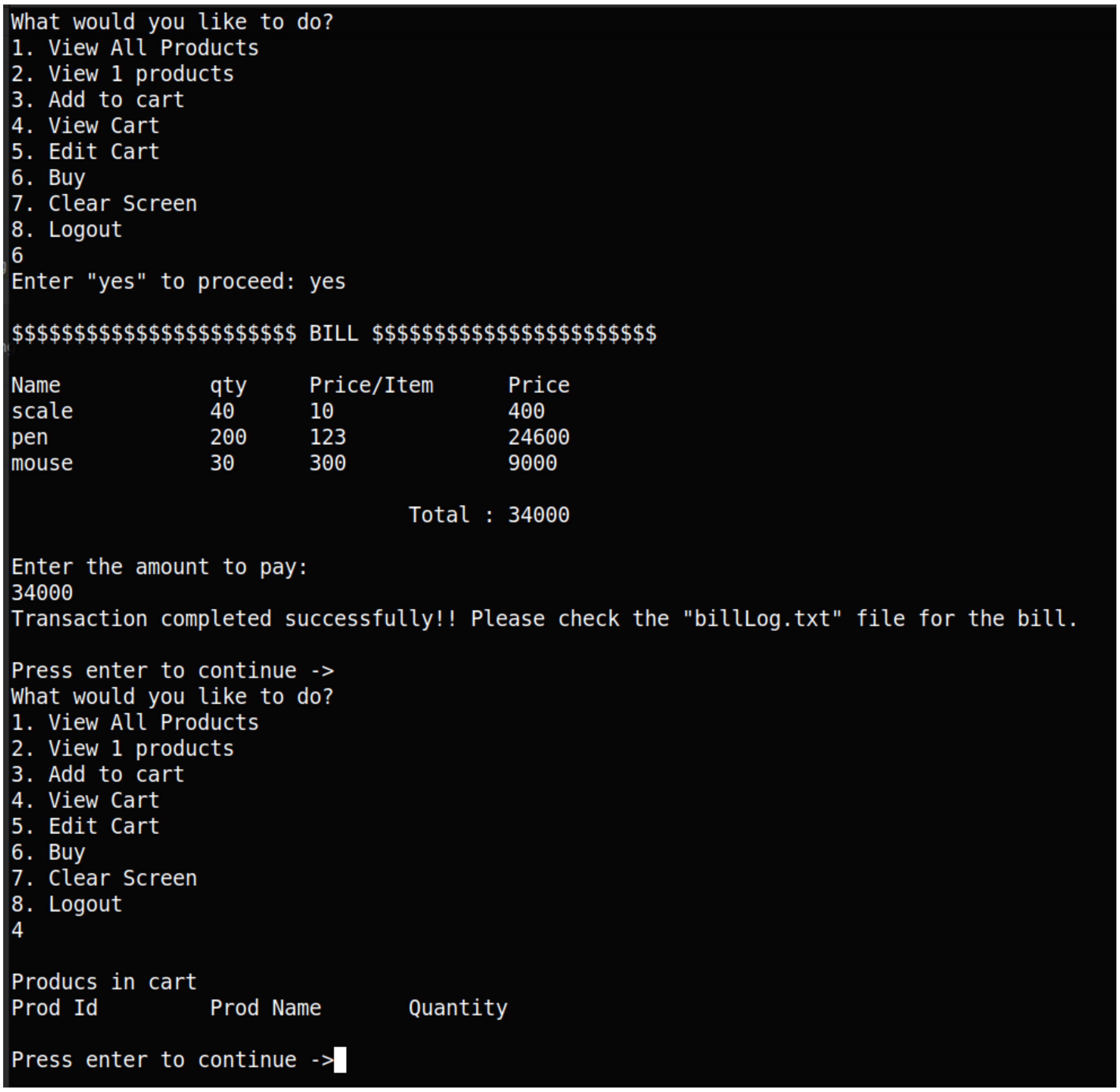
```
What would you like to do?
1. View All Products
2. View 1 products
3. Add to cart
4. View Cart
5. Edit Cart
6. Buy
7. Clear Screen
8. Logout
3
Enter product id: 00000002
Enter product quantity: 200
Product added to cart
Press enter to continue ->
What would you like to do?
1. View All Products
2. View 1 products
3. Add to cart
4. View Cart
5. Edit Cart
6. Buy
7. Clear Screen
8. Logout
4

Producs in cart
Prod Id         Prod Name       Quantity
00000003        scale           40
00000002        pen             200
00000004        mouse           30

Press enter to continue ->
```

8. Buy



```
What would you like to do?
1. View All Products
2. View 1 products
3. Add to cart
4. View Cart
5. Edit Cart
6. Buy
7. Clear Screen
8. Logout
6
Enter "yes" to proceed: yes

$$$$$$$$$$$$$$$$$$$$$$$ BILL $$$$$$$$$$$$$$$$$$$$$$$

Name        qty     Price/Item      Price
scale       40      10              400
pen         200     123             24600
mouse       30      300             9000
                           Total :  34000

Enter the amount to pay: 123456
Transaction failed :(. Please try again

Press enter to continue ->
```

Transaction Failure



```
What would you like to do?
1. View All Products
2. View 1 products
3. Add to cart
4. View Cart
5. Edit Cart
6. Buy
7. Clear Screen
8. Logout
6
Enter "yes" to proceed: yes

$$$$$$$$$$$$$$$$$$$$$$$ BILL $$$$$$$$$$$$$$$$$$$$$$$

Name            qty     Price/Item      Price
scale           40      10              400
pen             200     123             24600
mouse           30      300             9000

                                Total : 34000

Enter the amount to pay:
34000
Transaction completed successfully!! Please check the "billLog.txt" file for the bill.

Press enter to continue ->
What would you like to do?
1. View All Products
2. View 1 products
3. Add to cart
4. View Cart
5. Edit Cart
6. Buy
7. Clear Screen
8. Logout
4

Producs in cart
Prod Id         Prod Name       Quantity

Press enter to continue ->
```

Transaction Success