**GROCERY MANAGEMENT**

*A*

*Mini Project Report*

*Submitted in partial fulfilment of the*

*Requirements for the award of the Degree of*

**BACHELOR OF ENGINEERING**

IN

**INFORMATION TECHNOLOGY**

By

**M. Abignya Reddy <1602-19-737-001 >**

**G.R. Freya Rachel <1602-19-737-032 >**

****

**Department of Information Technology**

**Vasavi College of Engineering (Autonomous)**

**(Affiliated to Osmania University)**

**Ibrahimbagh, Hyderabad-31**

**2020**

**VASAVI COLLEGE OF ENGINEERING(AUTONOMOUS)**

**(AFFILIATED TO OSMANIA UNIVERSITY)**

**HYDERABAD - 500 031**

**Department of Information Technology**

****

**DECLARATION BY THE CANDIDATE**

We,M. Abignya Reddy and G.R. Freya Rachel, bearing hall ticket numbers, <1602-19-737-001> and <1602-19-737-032>, hereby declare that the project report entitled GrocerDepartment of Information Technology, Vasavi College of Engineering, Hyderabad, is submitted in partial fulfilment of the requirement for the award of the degree of **Bachelor of Engineering** in **Information Technology**

This is a record of bonafide work carried out by me and the results embodied in this project report have not been submitted to any other university or institute for the award of any other degree or diploma.

<M. Abignya Reddy >

<1602-19-737-001>

< G.R. Freya Rachel>

< 1602-19-737-032>

(Faculty In-Charge)

(Head, Dept of IT)

**ACKNOWLEDGEMENT**

With immense please we, M. Abignya Reddy and G.R. Freya Rachel presenting “Grocery Management” mini project report as per the curriculum of B.E second semester. We wish to thank all the people who gave us unending support.

We express our profound thanks to our course coordinator Ms. Leelavathy. And all those who have indirectly guided and helped us in our project.

We also take this opportunity to express a deep sense of gratitude to our HOD

Dr.K.Ram Mohan Rao Sir whose valuable guidance and kind supervision given to us

throughout the course which shaped our present work.

M. Abignya Reddy

1602-19-737-001

G.R. Freya Rachel

1602-19-737-032

Mini Project -1

Title: Grocery Management

**ABSTRACT**

Our mini project is about grocery management, it basically makes the grocery store owner and the customers work easier. Especially for regular customers how buy the groceries every month. As you run the program two options pop up 1) Are you a customer and 2) Are you the grocery store owner. If you are a customer you need to enter your details and then you can select your groceries, after you are done, it will display your list of groceries with pricing and total bill amount, then you can ask to generate the final bill or continue shopping. If you are the grocery store owner you will have access to all the information given by your customer. It will password secured.

M. Abignya Reddy

1602-19-737-001

G.R. Freya Rachel

1602-19-737-032

Mini Project -1

Title: Grocery Management

**INTRODUCTION**

**AIM OF THE PROJECT**

Objective of proposed system, is to provide Online Grocery Shopping solution to consumers and vendors. It will automate some of the basic operations of an online store. Scope would be to provide basic functionalities using a web application so that those manual process can be automated. It will include to provide administration access to vendors and admins and user specific access to customers.

This project is aimed at developing a Web application that depicts online shopping of grocery and purchasing. Using this software, can improve the efficiency of their services of sales product online.

Program includes modules like:

* Register
* Login
* Selection of groceries
* Append the grocery list
* Track Customers details
* Update grocery items

M. Abignya Reddy

1602-19-737-001

G.R. Freya Rachel

1602-19-737-032

**TECHNOLOGY**

**a. Software Requirements**

Software requirements deal with defining software resource requirements and pre requisites that need to be installed on a computer to provide optimal functioning of an application. These requirements or pre-requisites are generally not included in the software installation package and need to be installed separately before the software is installed.

Software requirements for this project are:

Operating System: Windows 10

Development tools: Vim editor, GCC compiler

These software requirements helped us to write our code efficiently and easily.

**b. Hardware Requirements**

The most common set of requirements defined by any operating system or software application is the physical computer resources, also known as hardware. A hardware requirements list is always accompanied by a hardware compatibility list (HCL), especially in case of operating systems. An HCL lists tested, compatibility and sometimes incompatible hardware devices for a particular operating system or application.

Hardware requirements for present project are:

Processor: Intel Core i5

RAM: 8GB

M. Abignya Reddy

1602-19-737-001

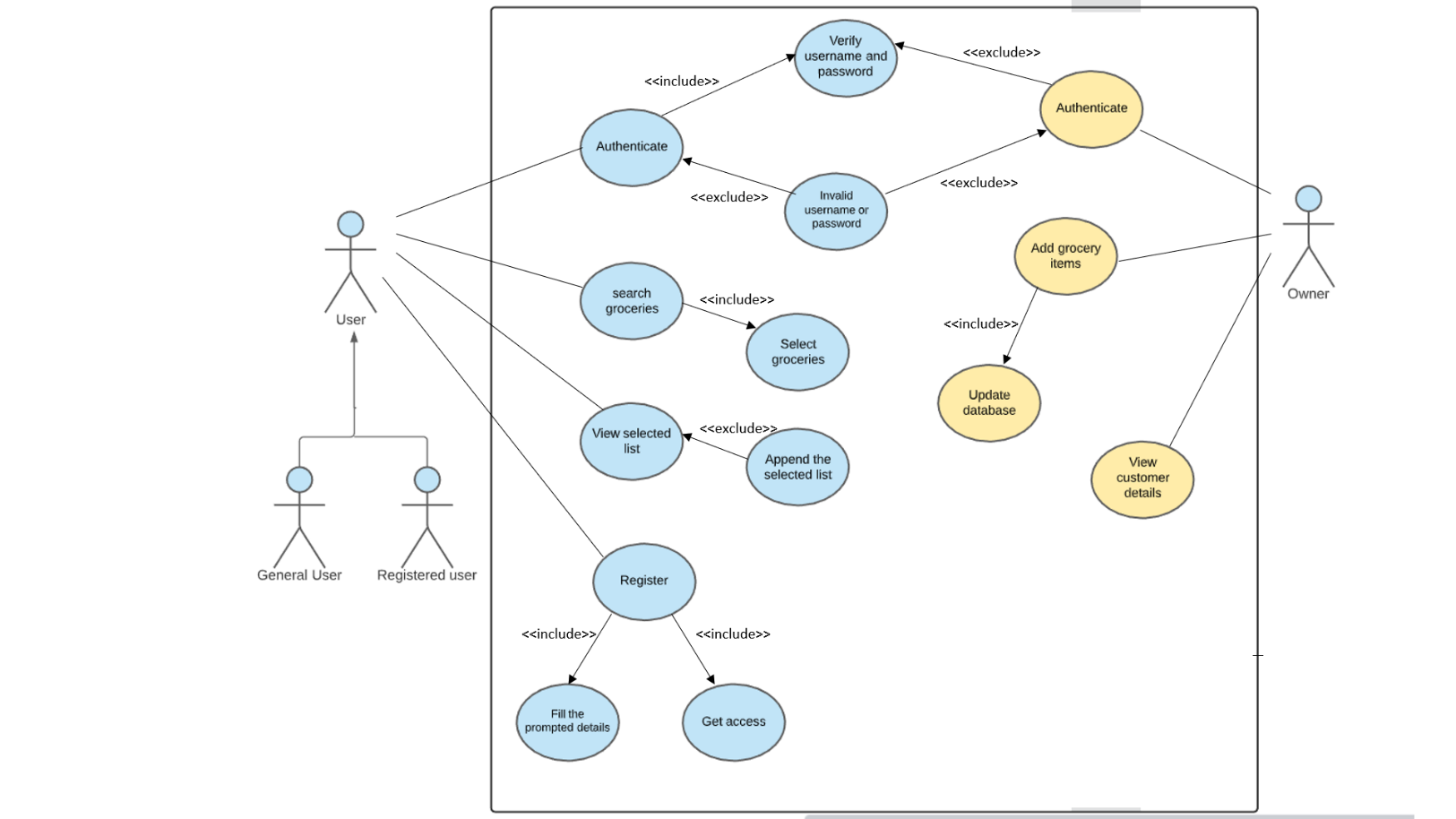
G.R. Freya Rachel

1602-19-737-032

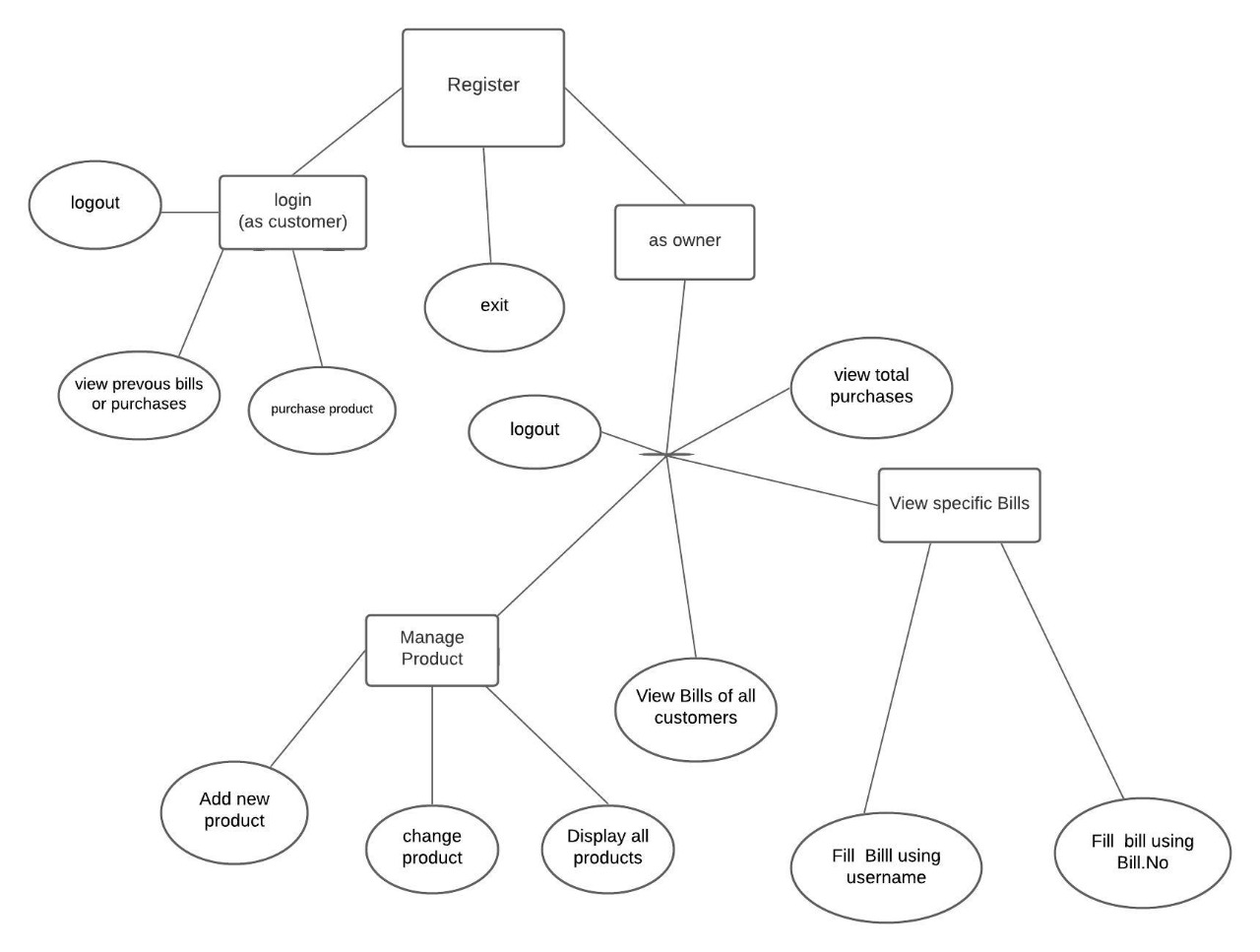
**PROPOSED WORK**

**a. Design**

**1. Use case diagram**

****

**2. Flowchart**

****

**b. Implementation**

**i. Description of main modules with respective results**

**1. Register**

**Description:** Allows new user to register for an account

void registration()

{

FILE \*fp,\*fp1;

char c,username[30];

int z=0;

fp=fopen("Web\_reg.txt","ab");

fp1=fopen("Web\_reg.txt","rb");

D:

printf("\n\n\t\t\t\tWELCOME TO REGISTRATION ZONE");

printf("\n\t\t\t\t----------------------------");

for(i=0;i<100;i++)

{

printf("\n\n\t\t\t\t ENTER USERNAME: ");

scanf("%s",username);

while(!feof(fp1))

{

fread(&w[i],sizeof(w[i]),1,fp1);

if(strcmp(username,w[i].name)==0)

{

printf("\n\n\t\t\tTHIS USERNAME ALREDY EXISTS");

getch();

goto D;

}

}

strcpy(w[i].name,username);

fclose(fp1);

printf("\n\n\t\t\t\t ENTER DESIRED PASSWORD: ");

while((c=getch())!=13)

{

w[i].pass[z++]=c;

printf("%c",c);

}

fwrite(&w[i],sizeof(w[i]),1,fp);

fclose(fp);

printf("\n\n\tPress enter if you agree with your Username and Password");

if((c=getch())==13)

{

clrscr();

printf("\n\n\t\tYour registration is completed successfully");

printf("\n\n\t\tDo you want to login into your account ?\n\n\t\t ");

printf("> PRESS 1 IF 'YES'\n\n\t\t > PRESS 2 IF 'NO'\n\n\t\t\t\t");

scanf("%d",&n);

switch(n)

{

case 1:

{

clrscr();

login();

break;

}

case 2:

{

clrscr();

printf("\n\n\n\t\t\t\t\tTHANK YOU FOR REGISTERING");

break;

}

}

break;

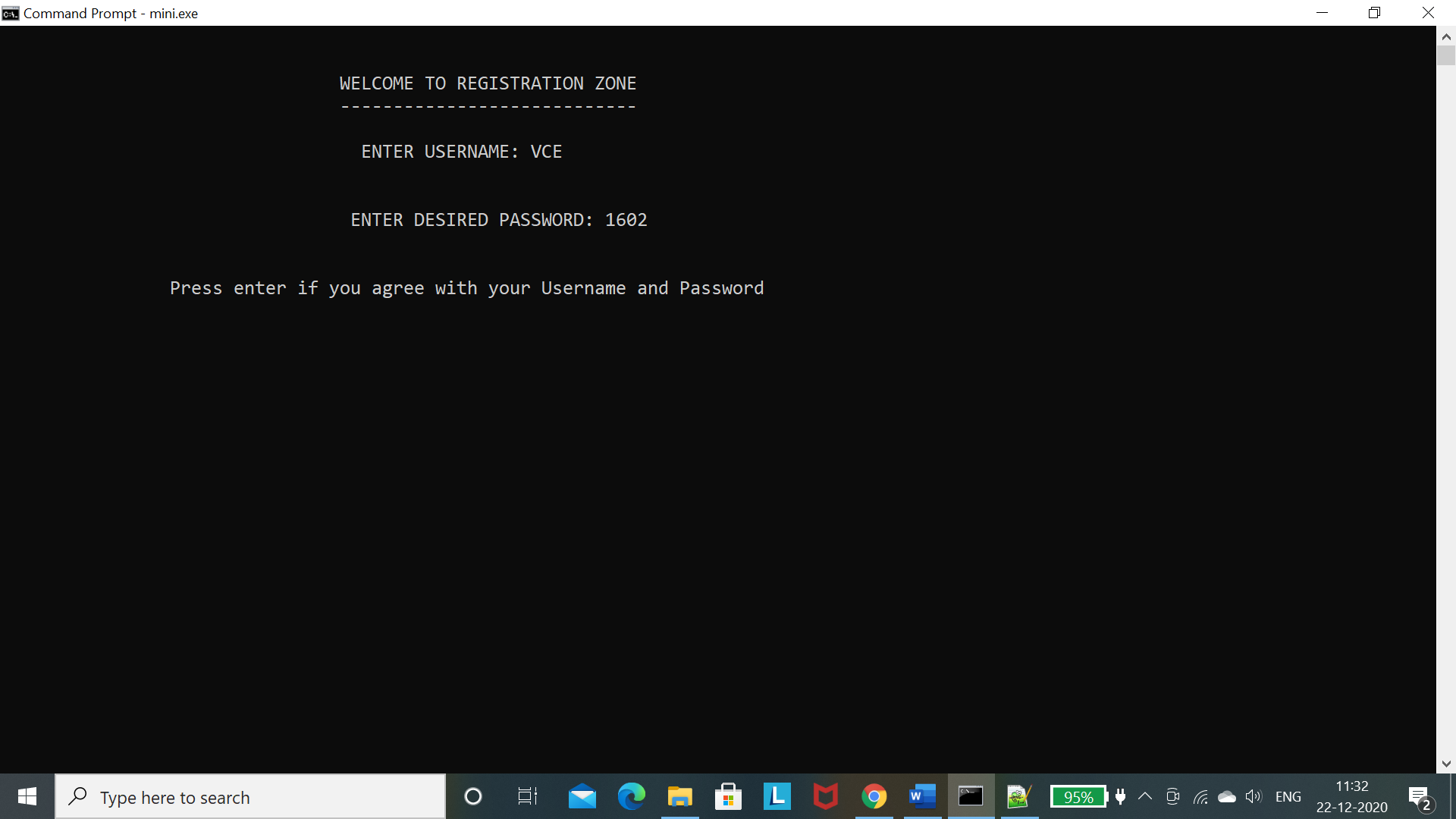
}

break;

}

getch();

}



**2. Customer Login**

**Description:** Allows registered users to Login and displays menu regarding shopping.

void login()

{

FILE \*fp;

char c,name[30],pass[30];

int z=0,ch;

int checku,checkp;

fp=fopen("Web\_reg.txt","rb");

printf("\n\n\t\t\t\tWELCOME TO LOGIN ZONE");

printf("\n\t\t\t\t---------------------");

for(i=0;i<1000;i++)

{

printf("\n\n\t\t\t\t ENTER USERNAME: ");

scanf("%s",name);

printf("\n\n\t\t\t\t ENTER PASSWORD: ");

while((c=getch())!=13)

{

pass[z++]=c;

printf("%c",'\*');

}

pass[z]='\0';

if(fp==NULL)

{

printf("\n\n\n\t\t\tYou are not a Registered User\n\n \t\t\tPress enter to register yourself");

if(getch()==13)

clrscr();

registration();

}

while(!feof(fp))

{

fread(&w[i],sizeof(w[i]),1,fp);

checku=strcmp(name,w[i].name);

checkp=strcmp(pass,w[i].pass);

if(checku==0&&checkp==0)

{

clrscr();

printf("\n\n\n\t\t\tYOU HAVE LOGGED IN SUCCESSFULLY!!");

getch();

while(1)

{

Y:

clrscr();

gotoxy(50,5);

printf("Welcome !! Select one option");

gotoxy(50,7);

printf("----------------------------");

gotoxy(50,10);

printf("1.Purchase new products");

gotoxy(50,13);

printf("2.View previous purchases(bills)");

gotoxy(50,16);

printf("3.Log Out");

gotoxy(53,20);

printf("Enter your choice : ");

scanf("%d",&ch);

switch(ch)

{

case 1: purchaseProduct(name);break;

case 2: displayPreviousBills(name);break;

case 3: exit(0);

default:

{

gotoxy(50,24);

printf("INVALID CHOICE");

gotoxy(50,27);

printf("Click on Enter to re-Enter your choice...");

if(getch()==13)

clrscr();

goto Y;

}

}

}

break;

}

else if(checku==0&&checkp!=0)

{

printf("\n\n\n\t\t\tWRONG PASSWORD!!");

printf("\n\n\t\t\t\t (Press 'Y' to re-login)");

if(getch()=='y'||getch()=='Y')

login();

break;

}

}

if(feof(fp))

{

printf("\n\n\n\t\t\tYou are not a Registered User\n\n \t\t\tPress enter to register yourself");

if(getch()==13)

clrscr();

registration();

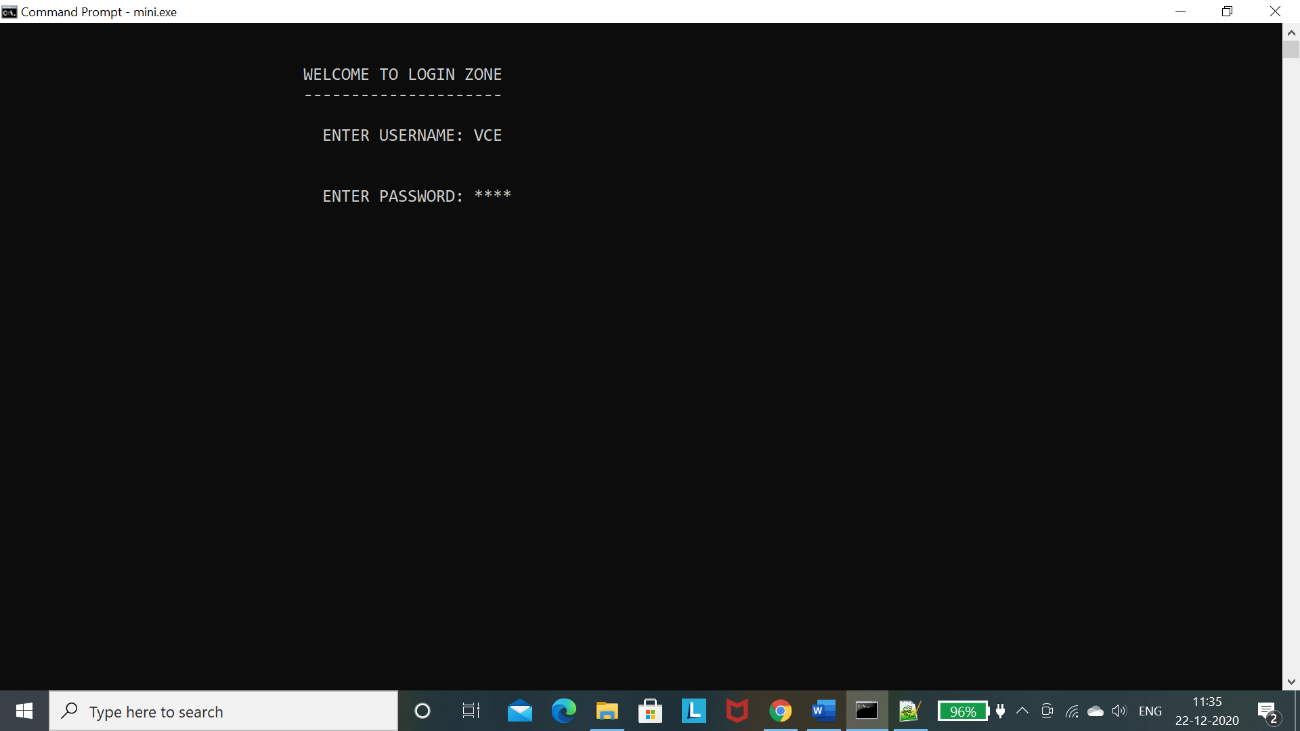
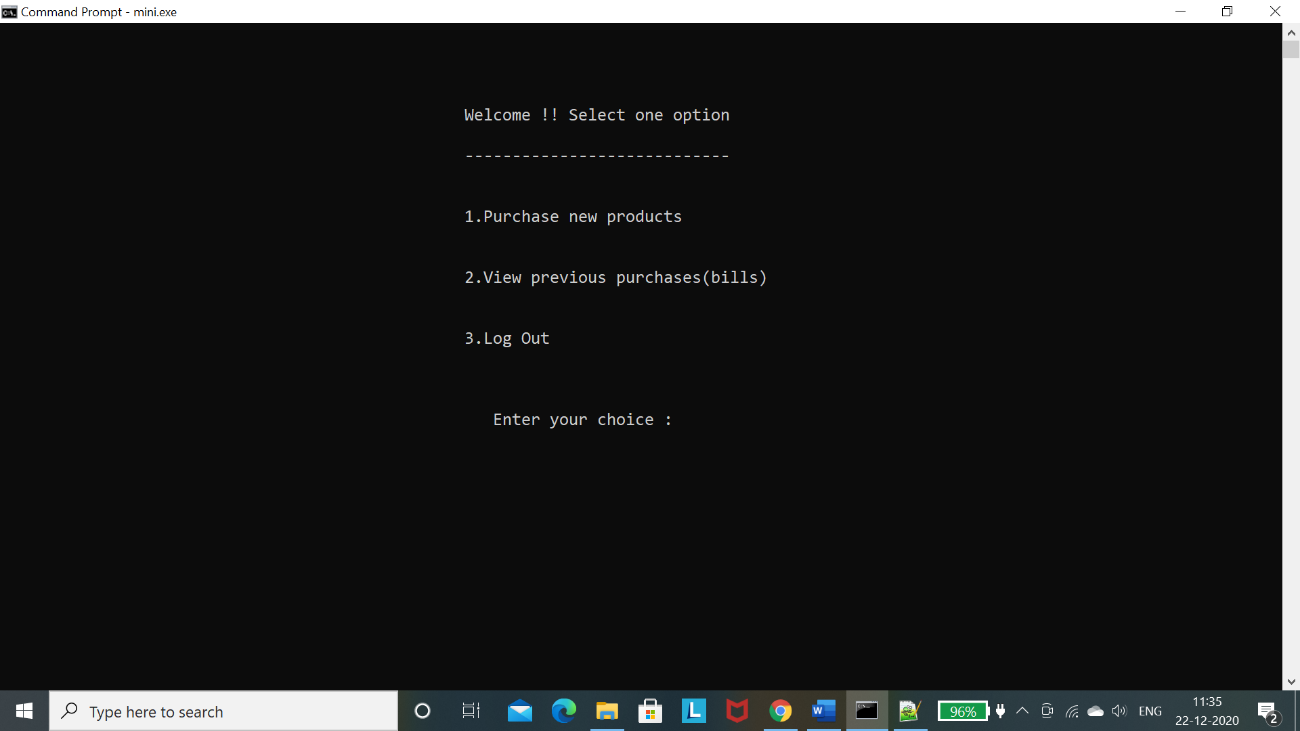
}

break;

}

getch();

}

**3. Owner Login**

**Description:** Allows owner to Login and displays menu for customer and product details.

void OwnerLogin()

{

gotoxy(50,7);

printf("WELCOME TO LOGIN ZONE");

gotoxy(50,8);

printf("---------------------");

char pass[4];

char pin[]="1234";

int x,y=3,ch;

for(x=1;x<5;x++)

{

gotoxy(50,12);

printf("ENTER 4 DIGIT PIN: ");

for(int p=0;p<4;p++)

{

pass[p]=getch();

printf("\*");

}

pass[4]='\0';

getch();

if (!strcmp(pass,pin))

{

clrscr();

gotoxy(50,12);

printf("YOU HAVE LOGGED IN SUCCESSFULLY!!");

gotoxy(50,17);

printf("Enter any key to cont...");

x=0;

getch();

clrscr();

while(1)

{

Z:

clrscr();

gotoxy(50,5);

printf("CUSTOMER AND PRODUCT DETAILS");

gotoxy(50,6);

printf("----------------------------");

gotoxy(50,10);

printf("1.Manage Products");

gotoxy(50,13);

printf("2.View Bills of All Customers");

gotoxy(50,16);

printf("3.View specific Bill");

gotoxy(50,19);

printf("4.View Total Purchases");

gotoxy(50,22);

printf("5.Log Out");

gotoxy(53,26);

printf("Enter your choice : ");

scanf("%d",&ch);

switch(ch)

{

case 1: manageProduct();break;

case 2: customerDetails();break;

case 4: totalPurchases();break;

case 3: specificBill();break;

case 5: exit(0);

default:

{

gotoxy(50,30);

printf("INVALID CHOICE");

gotoxy(50,33);

printf("Click on Enter to re-Enter your choice...");

if(getch()==13)

clrscr();

goto Z;

}

}

}

break;

}

else

{

clrscr();

gotoxy(50,2);

printf("WRONG PASSWORD!! %d more chances",y);

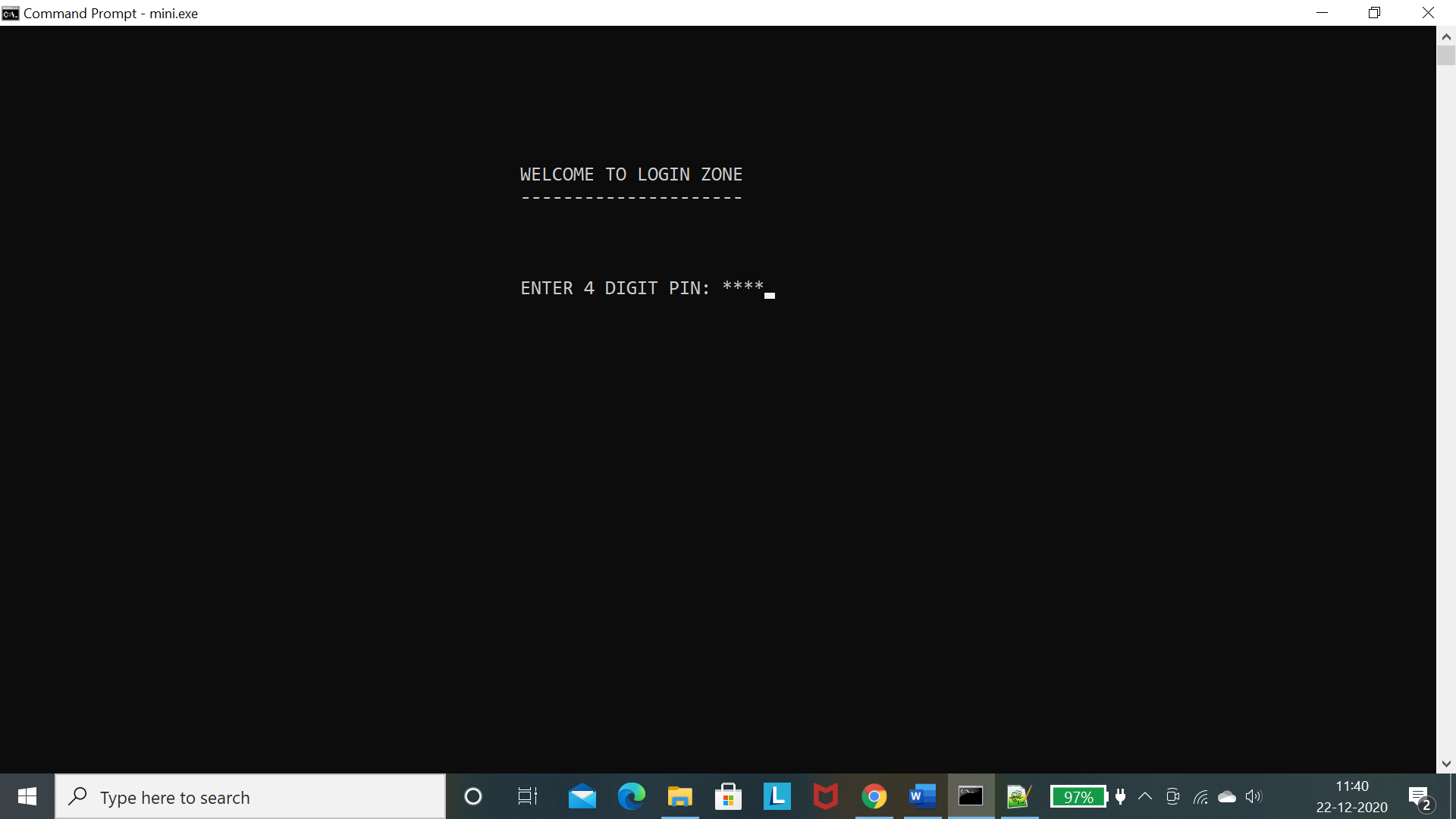
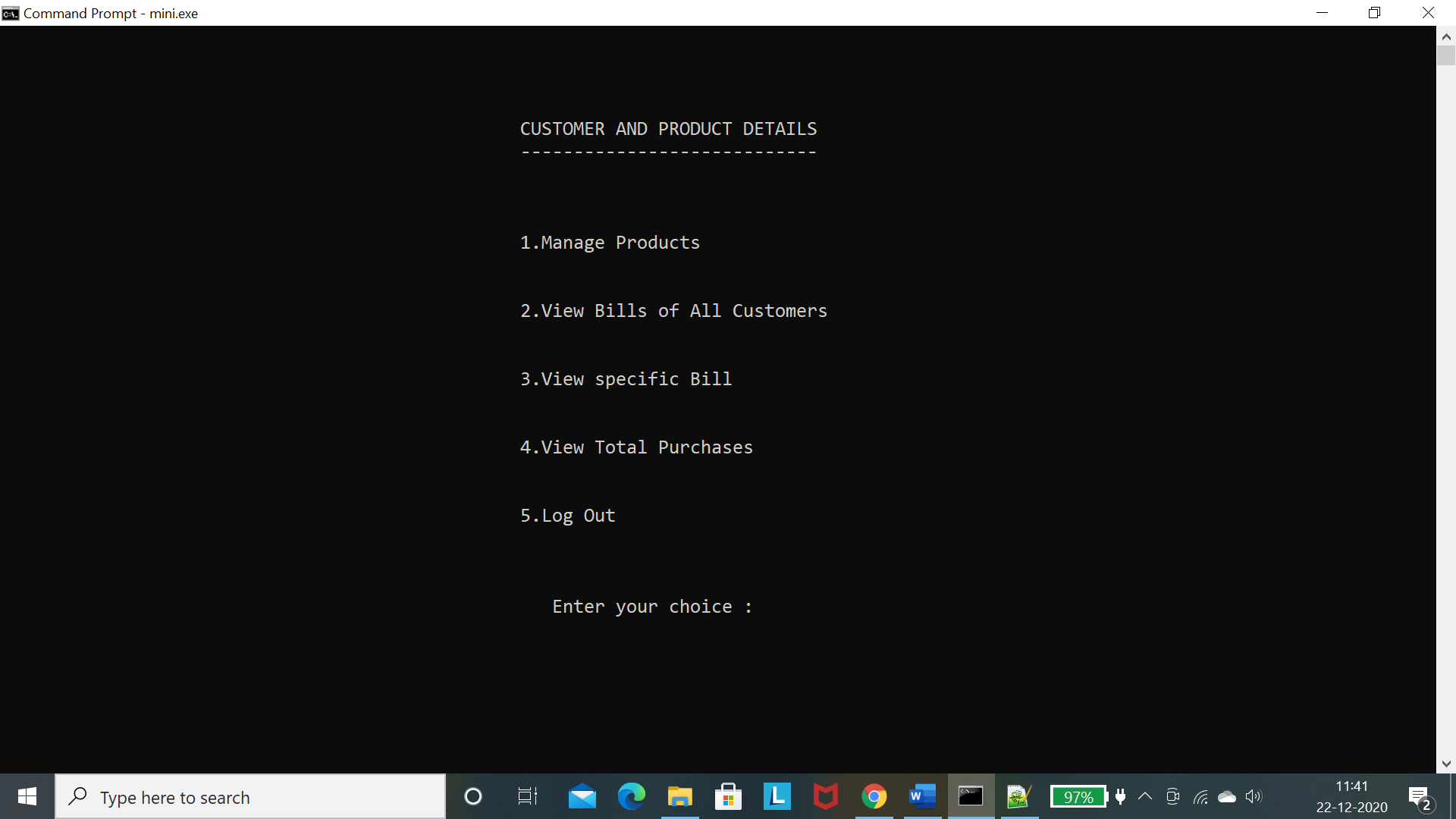
y--;

}

}

getch();

}

**4. Manage products**

**Description:** Displays menu related to managing the products.

void manageProduct()

{

int ch;

int back=0;

while(1)

{

A:

clrscr();

gotoxy(50,5);

printf("Welcome to Product Management System");

gotoxy(50,7);

printf("------------------------------------");

gotoxy(50,10);

printf("1.Add New Product");

gotoxy(50,13);

printf("2.Display All Products");

gotoxy(50,16);

printf("3.Change Product Quantity");

gotoxy(50,19);

printf("4.Back");

gotoxy(53,23);

printf("Enter your choice : ");

scanf("%d",&ch);

switch(ch)

{

case 1: addProduct();break;

case 2: displayAllProducts();break;

case 3: appendQuantity();break;

case 4: back=1;break;

default:

{

gotoxy(50,24);

printf("INVALID CHOICE");

gotoxy(50,27);

printf("Click on Enter to re-Enter your choice...");

if(getch()==13)

clrscr();

goto A;

}

}

if(back==1)

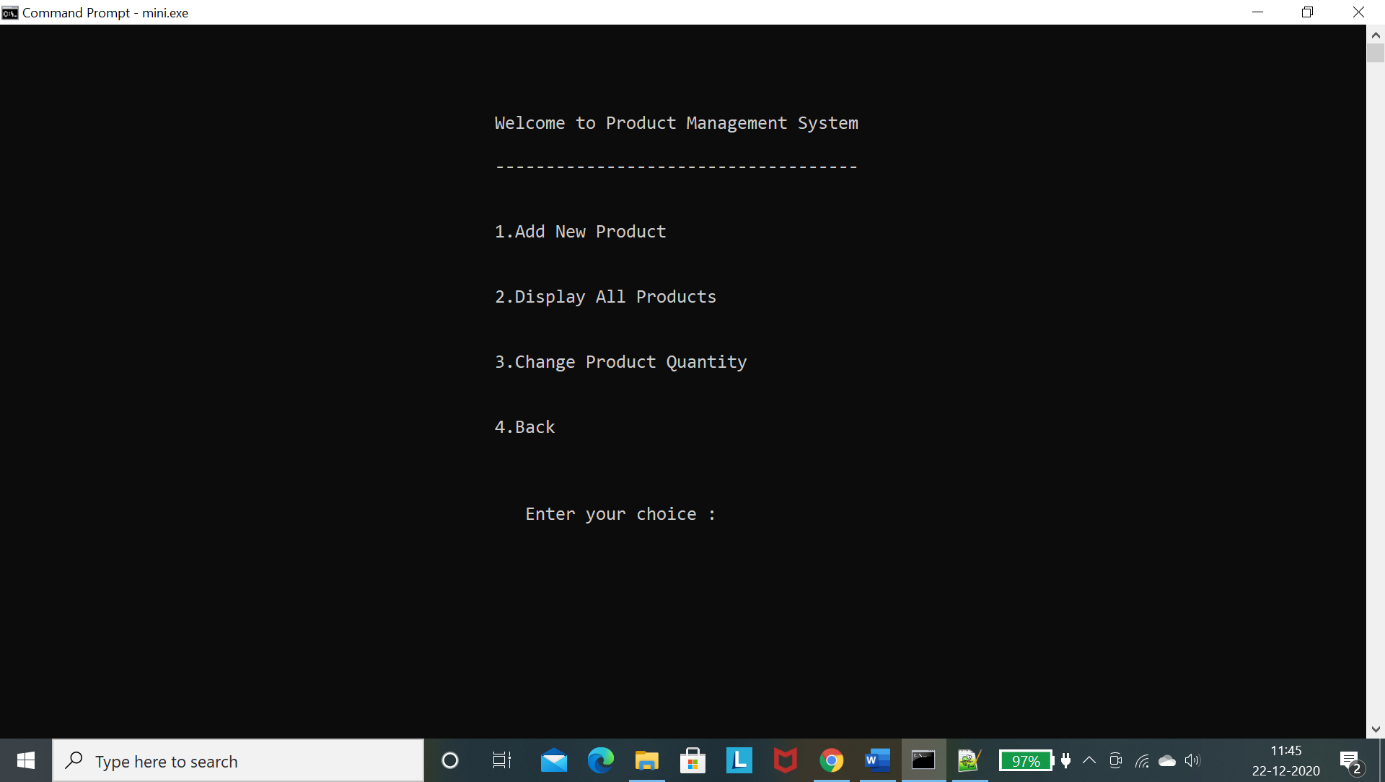
break;

else

continue;

}

}



**5. Add Products**

**Description:** This module helps the owner to add new products to display.

void addProduct()

{

FILE \*fp;

struct Product t1;

fp=fopen(fname,"ab");

clrscr();

t1.id=getID();

gotoxy(40,5);

printf("--------Enter details of the product to be added--------");

gotoxy(40,8);

printf("Enter product name : ");

scanf("%s",t1.name);

gotoxy(40,11);

printf("Enter product price>0 : ");

PQ:

scanf("%d",&t1.price);

if(t1.price<=0)

{

printf(" -Enter price>'zero' : ");

goto PQ;

}

gotoxy(40,14);

printf("Enter available quantity>0 : ");

MN:

scanf("%d",&t1.qty);

if(t1.qty<=0)

{

printf(" -Enter quantity>'zero' : ");

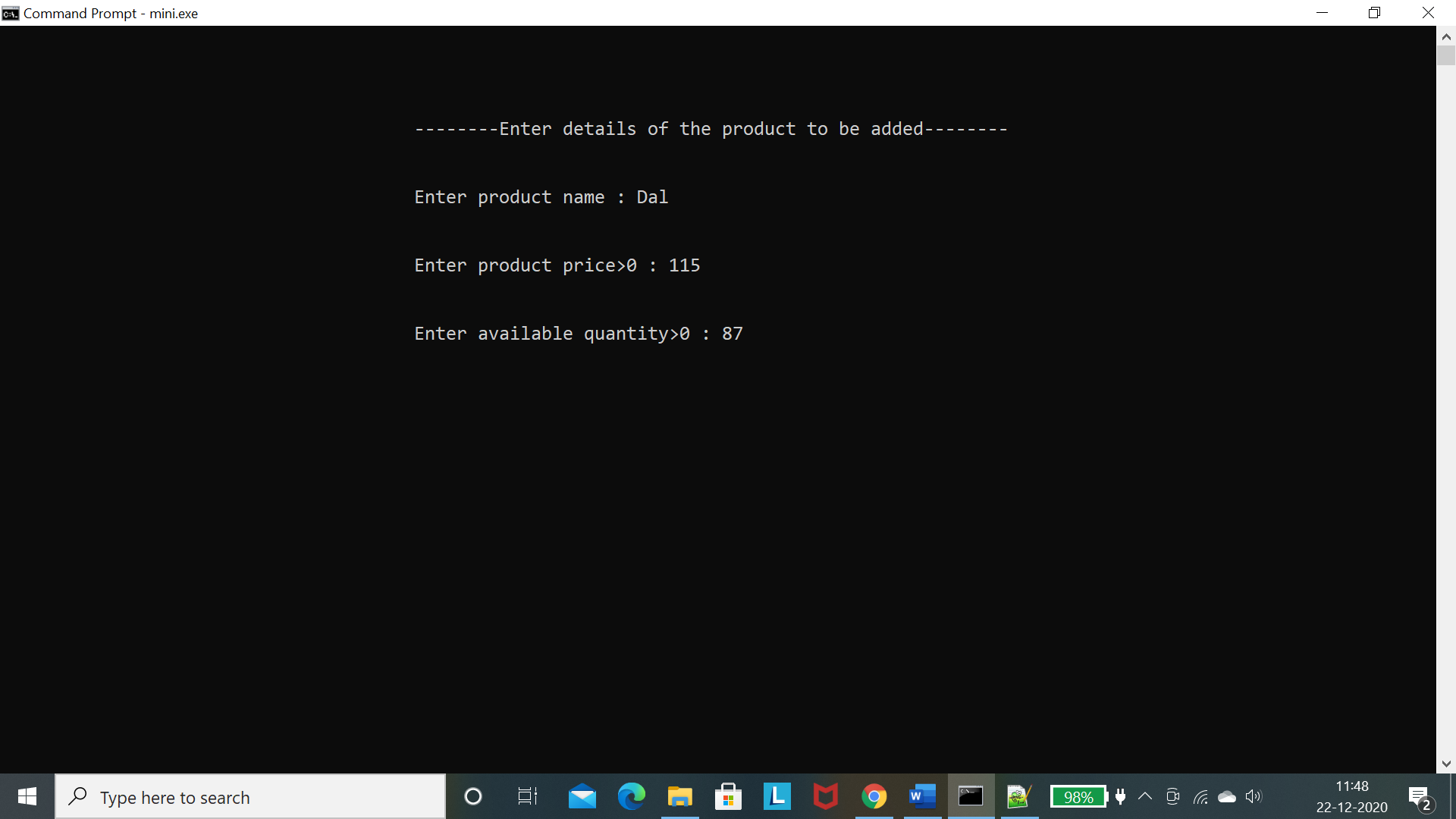
goto MN;

}

fwrite(&t1,sizeof(t1),1,fp);

fclose(fp);

}



**6. Display All Products**

**Description:** Allows the owner to view all the products on display.

void displayAllProducts()

{

FILE \*fp;

struct Product t;

clrscr();

fp=fopen(fname,"rb");

gotoxy(30,5);

printf("========================================================");

gotoxy(30,7);

printf("\t\t\t List Of All Products");

gotoxy(30,9);

printf("========================================================");

gotoxy(30,12);

printf("ID");

gotoxy(40,12);

printf("Name");

gotoxy(60,12);

printf("Quantity");

gotoxy(75,12);

printf("Price");

int a=15;

while(1)

{

fread(&t,sizeof(t),1,fp);

if(feof(fp))

{

break;

}

gotoxy(30,a);

printf("%d",t.id);

gotoxy(40,a);

printf("%s",t.name);

gotoxy(60,a);

printf("%d",t.qty);

gotoxy(75,a);

printf("%d",t.price);

a=a+2;

}

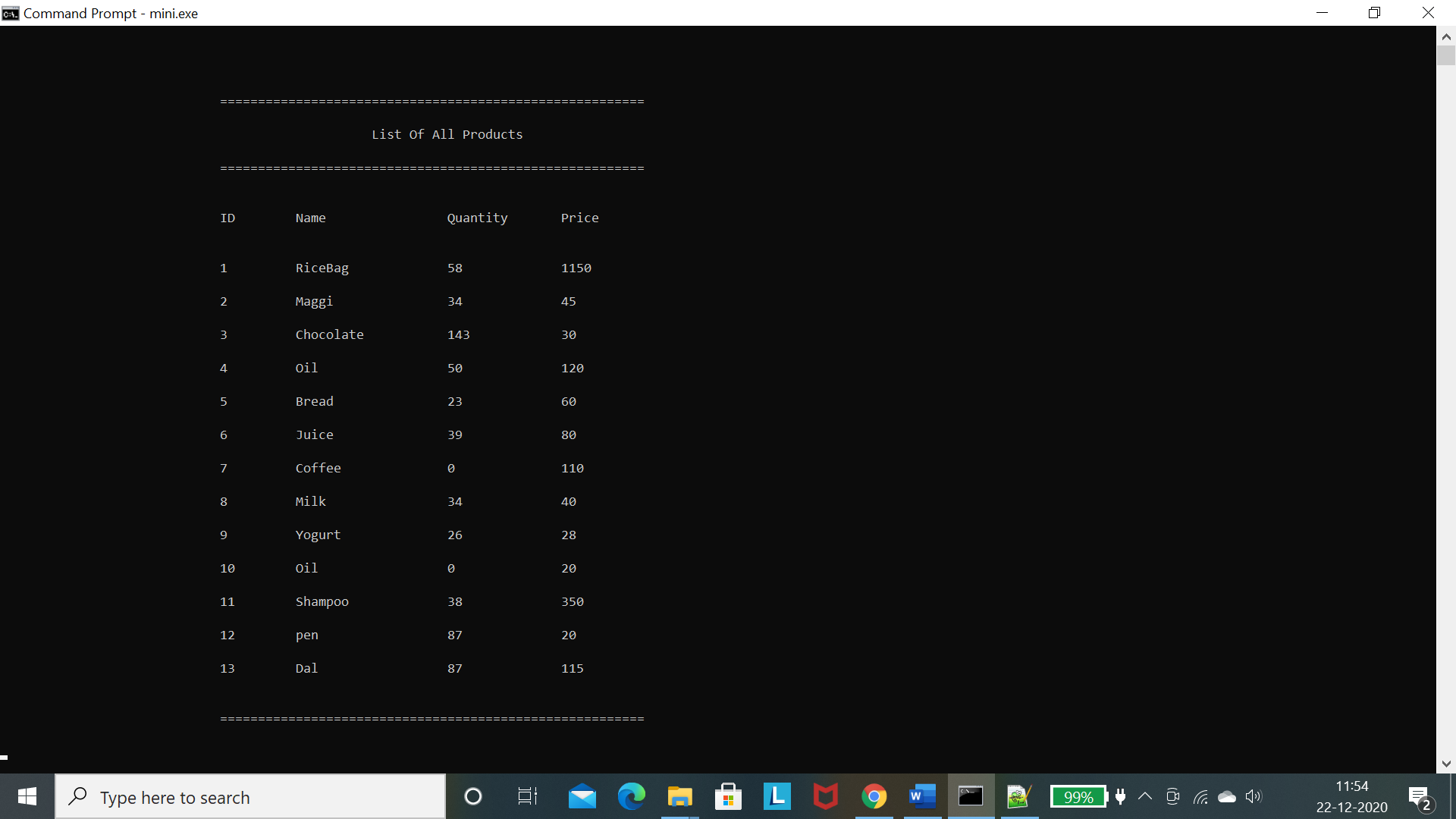
gotoxy(30,a+1);

printf("======================================== ================\n\n");

fclose(fp);

getch();

}



**7. Append Product Quantity**

**Description:** Allows the owner to append any product with the help of the product ID.

void appendQuantity()

{

FILE \*fp1;

G:

clrscr();

int n,x,lastId;

gotoxy(40,5);

printf("Enter product ID : ");

scanf("%d",&n);

fp1=fopen("id.txt","r");

fscanf(fp1,"%d",&lastId);

fclose(fp1);

if((lastId<n) || (n<=0))

{

gotoxy(35,10);

printf("-->> There is no product with that ID !!");

getch();

goto G;

}

FILE \*fp;

struct Product t;

fp=fopen(fname,"rb");

while(1)

{

fread(&t,sizeof(t),1,fp);

if(feof(fp))

{

break;

}

if(t.id==n)

{

gotoxy(40,7);

printf("Name of the product : %s",t.name);

gotoxy(40,9);

printf("Product quantity on display : %d",t.qty);

break;

}

}

fclose(fp);

gotoxy(40,12);

printf("Enter available product quantity : ");

scanf("%d",&x);

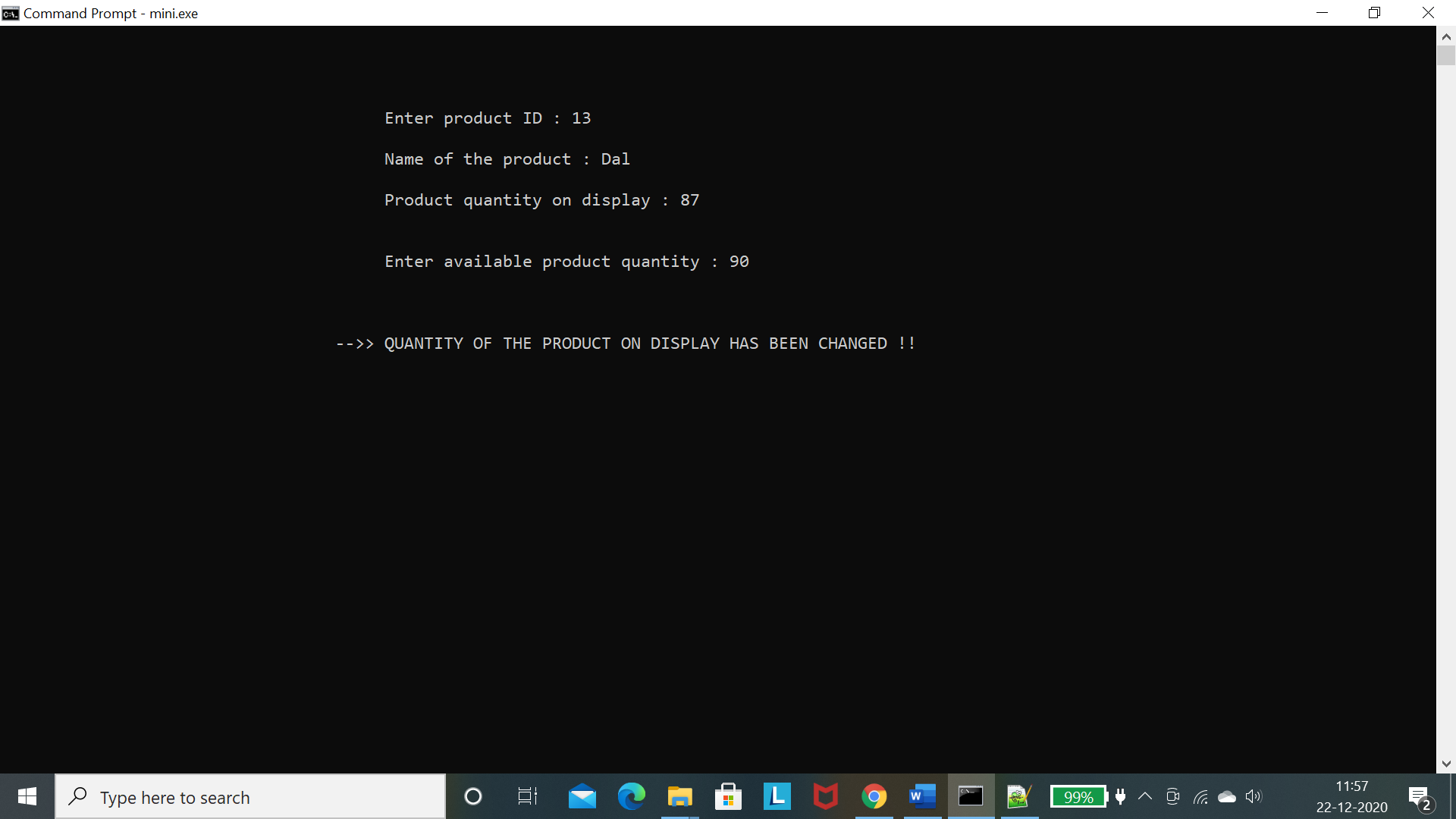
changeQuantity(n,x);

gotoxy(35,16);

printf("-->> QUANTITY OF THE PRODUCT ON DISPLAY HAS BEEN CHANGED !!");

getch();

}



**8. All Bills**

**Description:** Allows the owner to view bills of all customers from the very first bill to the last.

void allBills()

{

FILE \*fp;

struct Bill t;

char bill[20];

int value;

fp=fopen("billNo.txt","r");

fscanf(fp,"%d",&value);

fclose(fp);

for(int i=1;i<=value;i++)

{

clrscr();

sprintf(bill,"%s%d","",i);

strcat(bill,".dat");

fp=fopen(bill,"rb");

gotoxy(30,7);

printf("Bill no : %d",i);

gotoxy(30,12);

printf("ID");

gotoxy(40,12);

printf("Name");

gotoxy(60,12);

printf("Price");

int a=15;

while(1)

{

fread(&t,sizeof(t),1,fp);

if(feof(fp))

{

break;

}

gotoxy(30,9);

printf("Username : %s",t.user);

gotoxy(30,a);

printf("%d\t",t.pid);

gotoxy(40,a);

printf("%s\t",t.pname);

gotoxy(60,a);

printf("%d\t\n",t.pprice);

total=total+t.pprice;

a=a+2;

}

gotoxy(35,a+2);

printf("Total Bill Amount -->> Rs. %d ",total);

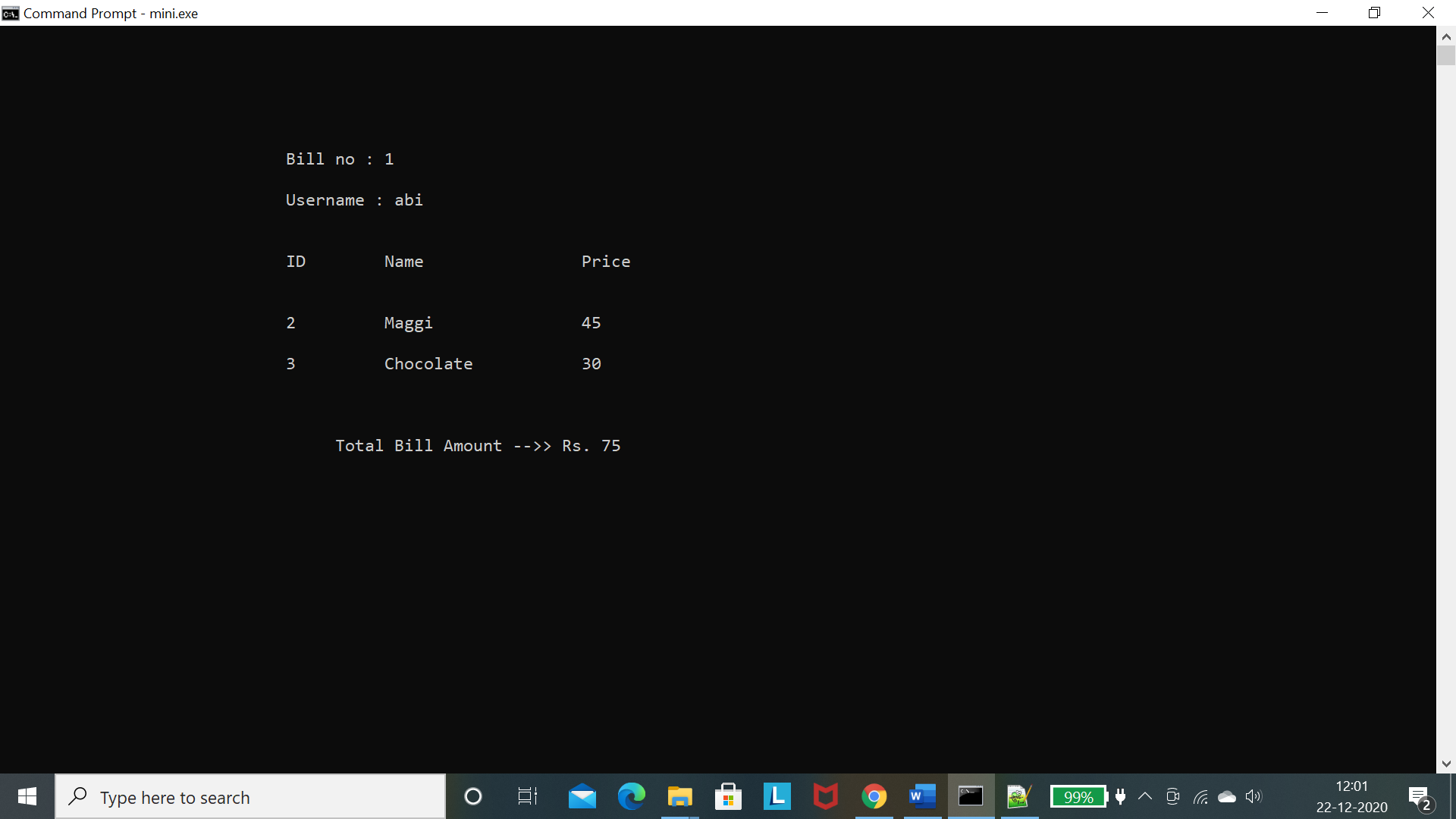
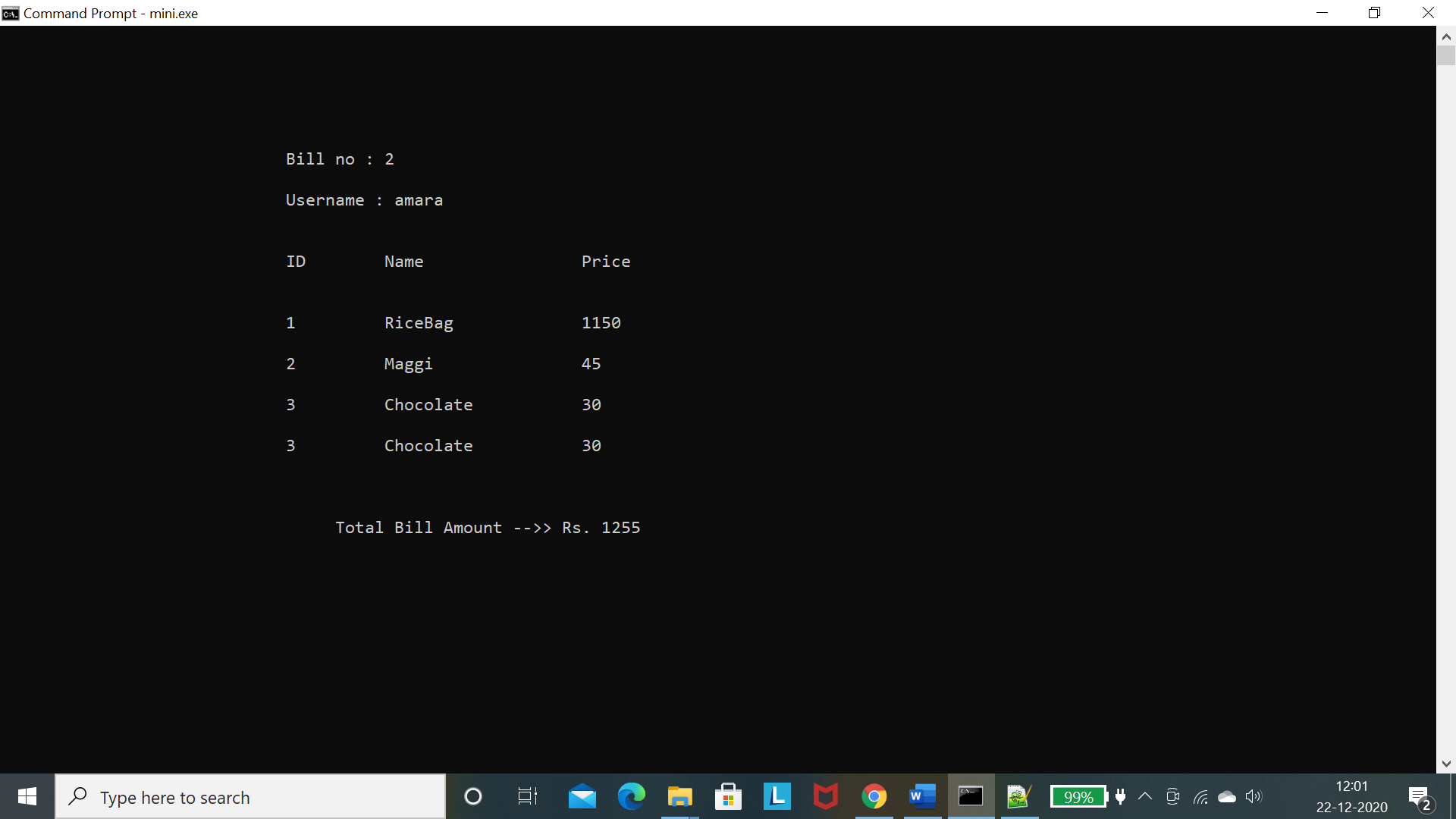
fclose(fp);

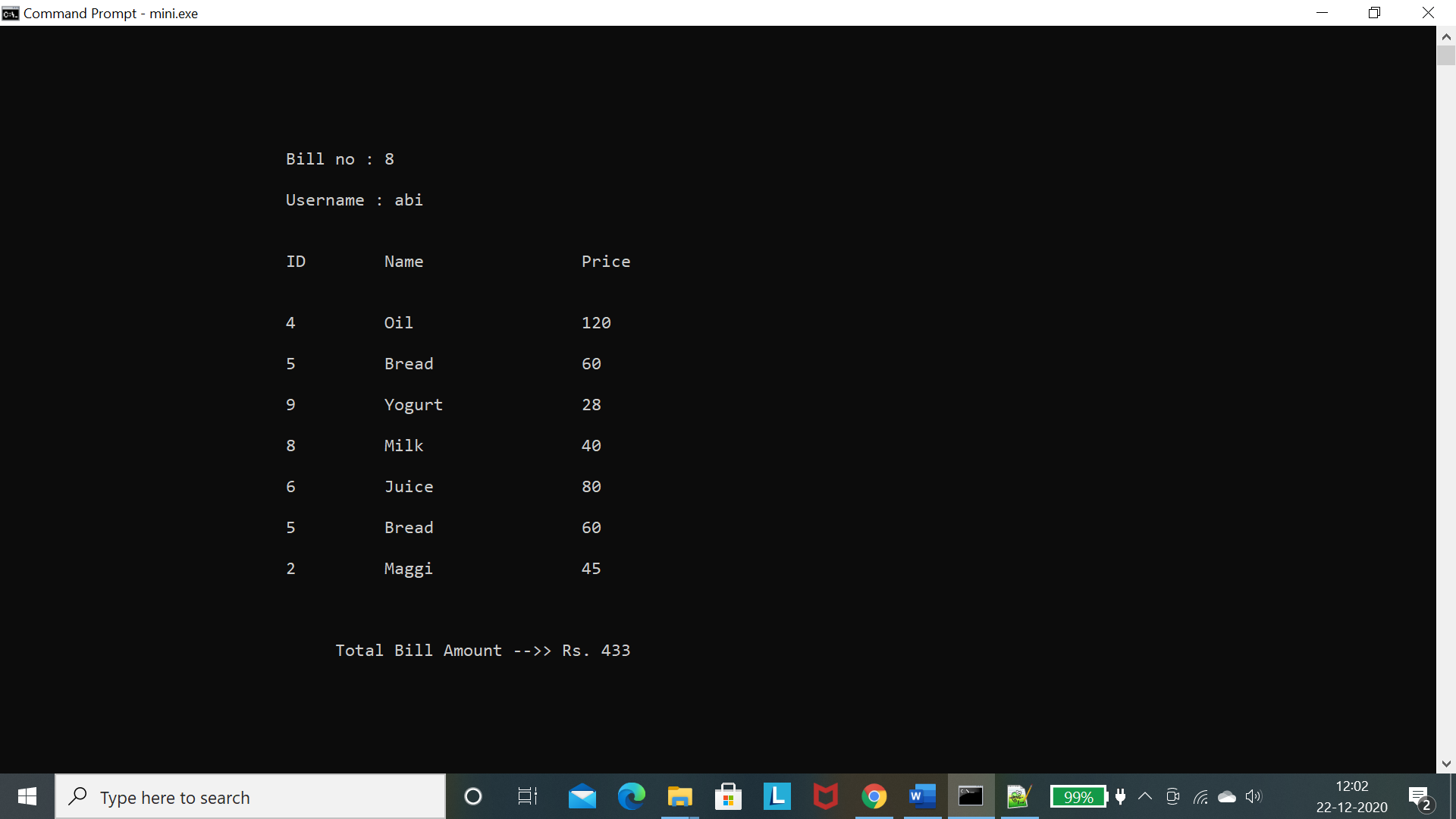
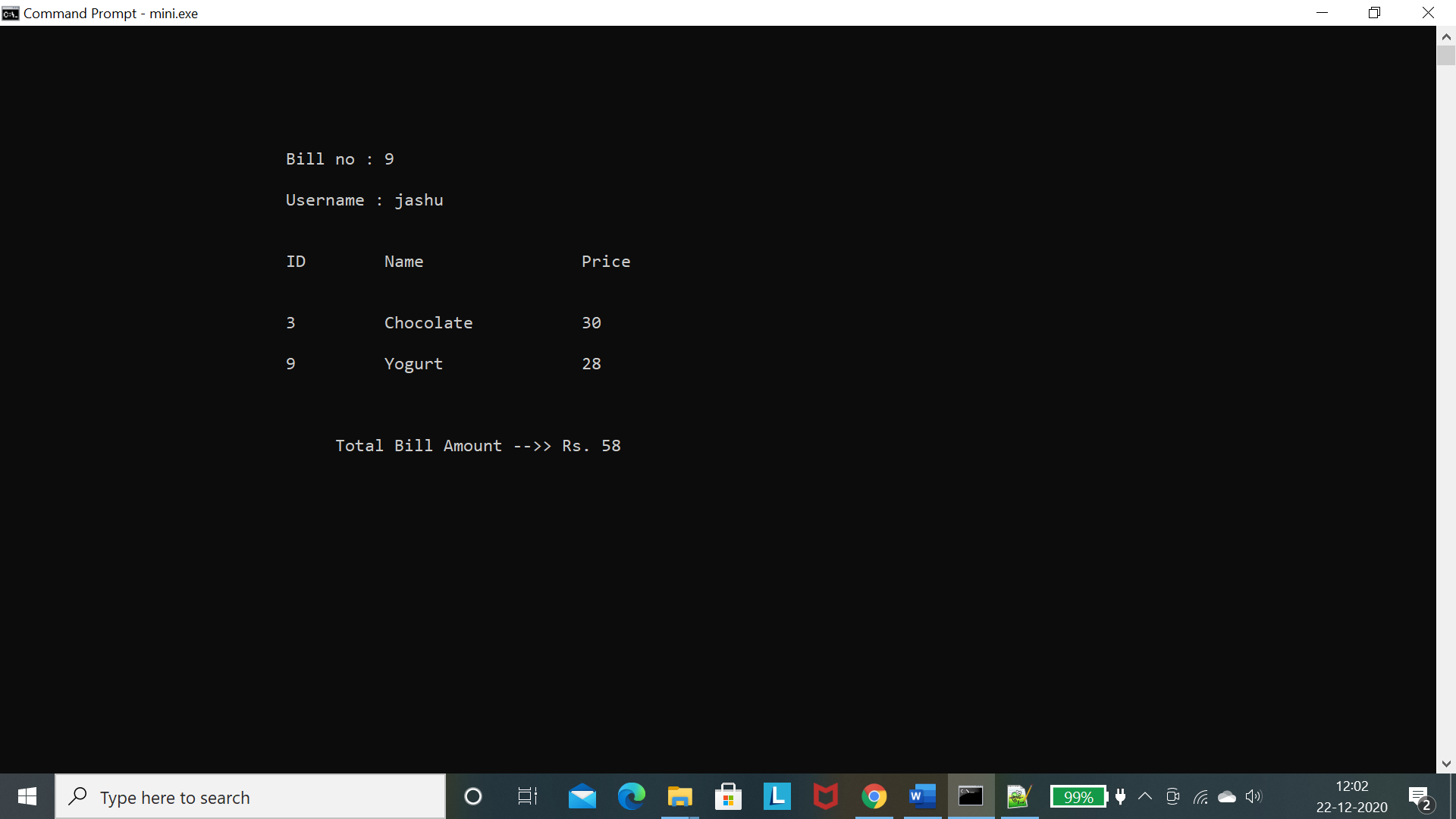
total=0;

getch();

}

}

  .3.4.5

.6.7. 

**9. Specific Bill**

**Description:** Allows the owner to view specific bill with the help of customer username or bill number.

void specificBill()

{

int ch,back=0;

while(1)

{

C:

clrscr();

gotoxy(40,10);

printf("Find Bill Using : 1.Username\t2.Bill Number\t3.Back");

gotoxy(40,13);

printf("Enter your choice : ");

scanf("%d",&ch);

switch(ch)

{

case 1: userBill();break;

case 2: numberBill();break;

case 3: back=1;break;

default:

{

gotoxy(40,17);

printf("INVALID CHOICE");

gotoxy(40,20);

printf("Click on Enter to re-Enter your choice...");

if(getch()==13)

clrscr();

goto C;}

}

if(back==1)

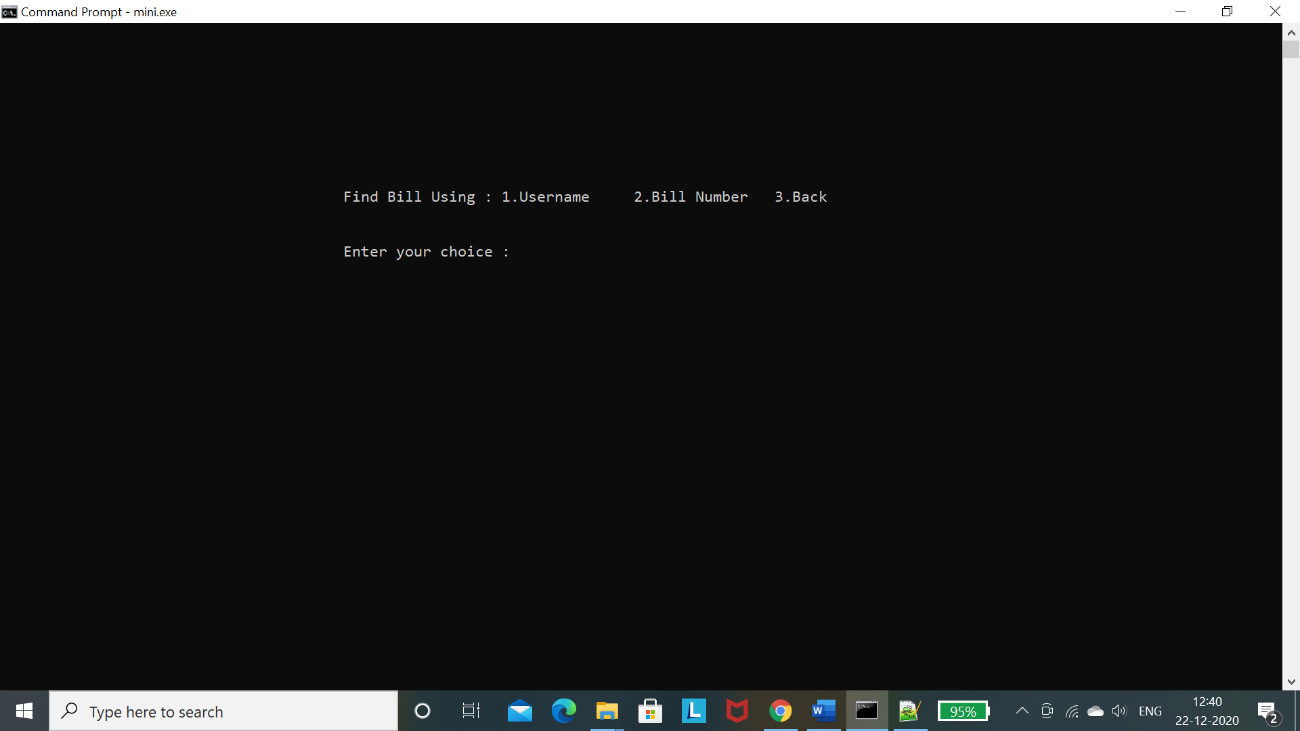
break;

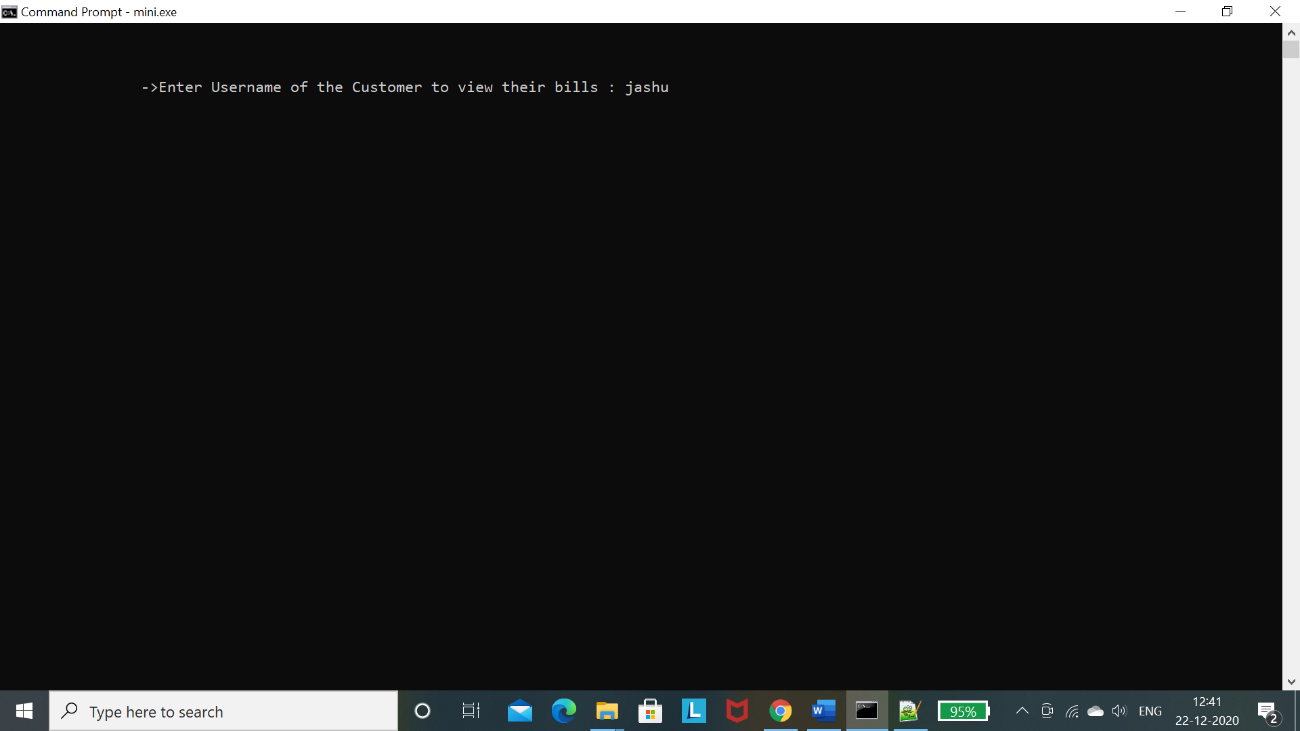
else

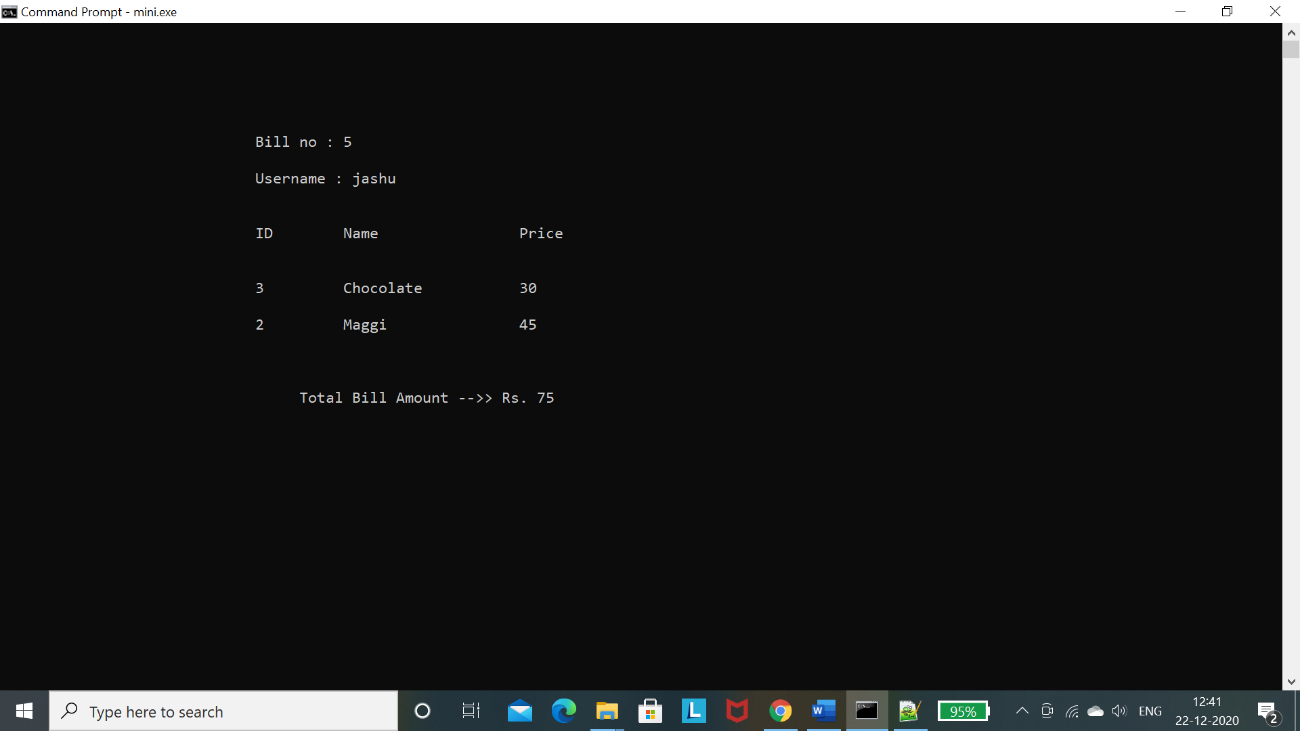
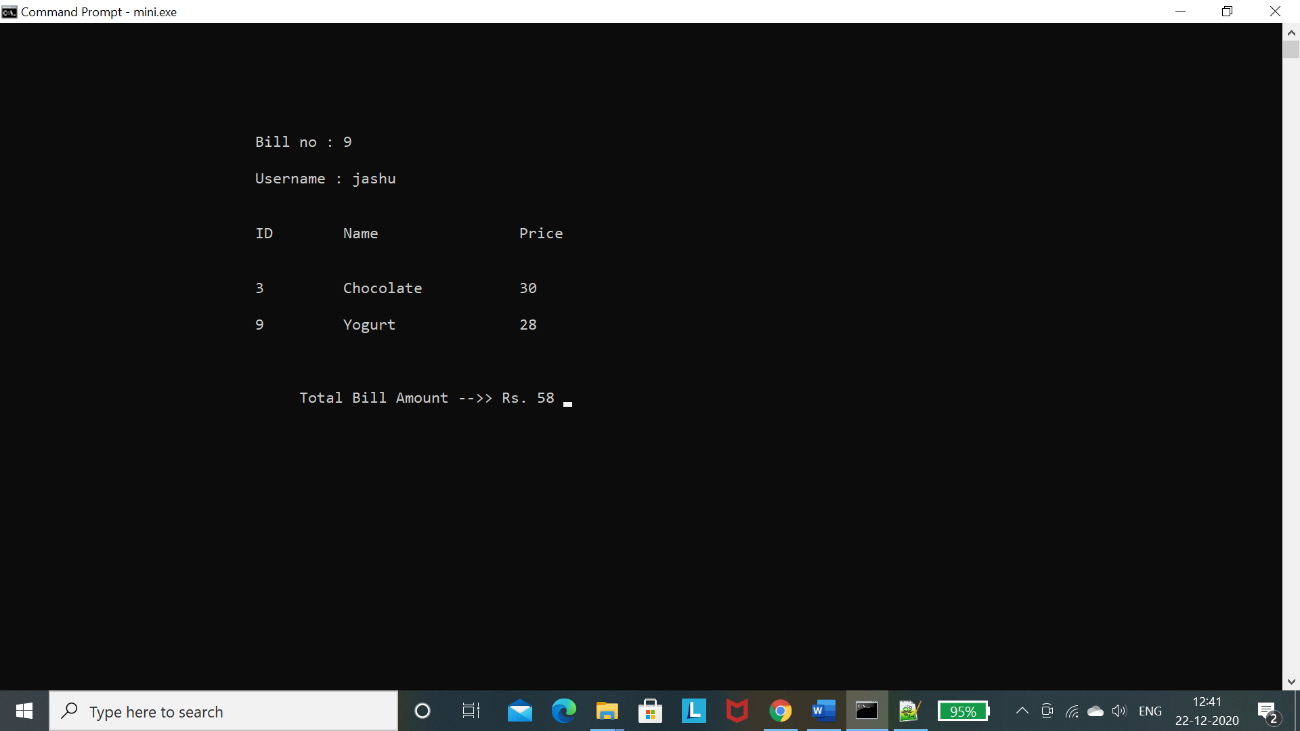
continue;

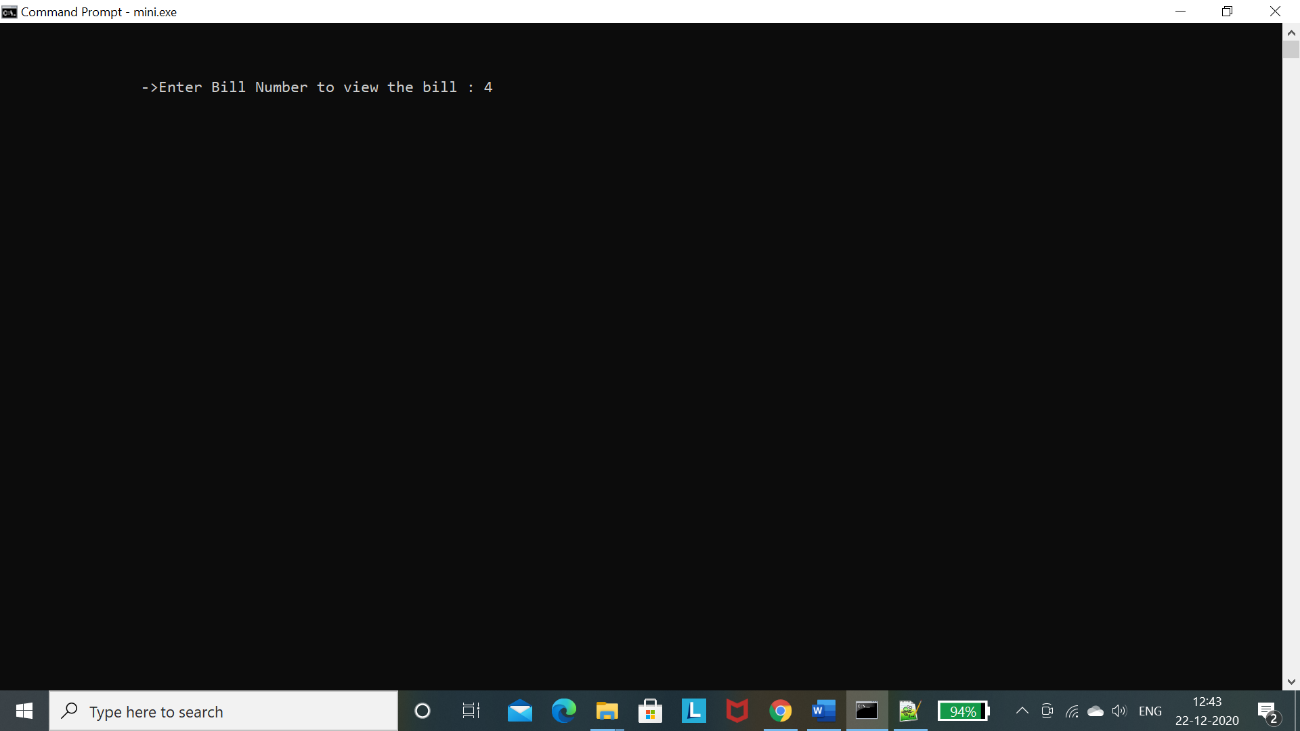
}

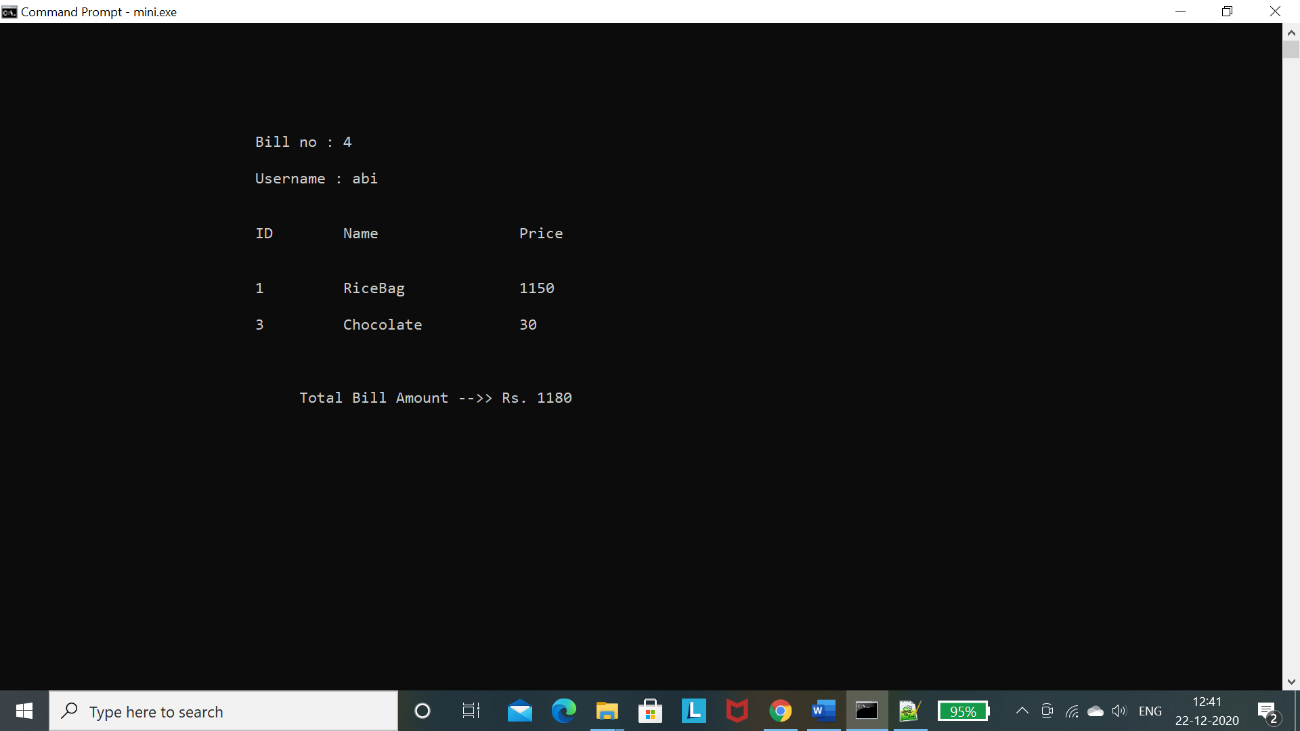
}



**1.** 

**2.** 



**10. Total Purchases**

**Description:** Allows the owner to view all purchased products with total amount.

void totalPurchases()

{

FILE \*fp;

struct Bill t;

clrscr();

fp=fopen(fAll,"rb");

gotoxy(30,5);

printf("========================================================");

gotoxy(30,7);

printf("\t\t\t Bill Details");

gotoxy(30,9);

printf("========================================================");

gotoxy(30,12);

printf("ID");

gotoxy(40,12);

printf("Name");

gotoxy(60,12);

printf("Price");

int a=15;

while(1)

{

fread(&t,sizeof(t),1,fp);

if(feof(fp))

{

break;

}

gotoxy(30,a);

printf("%d\t",t.pid);

gotoxy(40,a);

printf("%s\t",t.pname);

gotoxy(60,a);

printf("%d\t\n",t.pprice);

total=total+t.pprice;

a=a+2;

}

gotoxy(35,a+2);

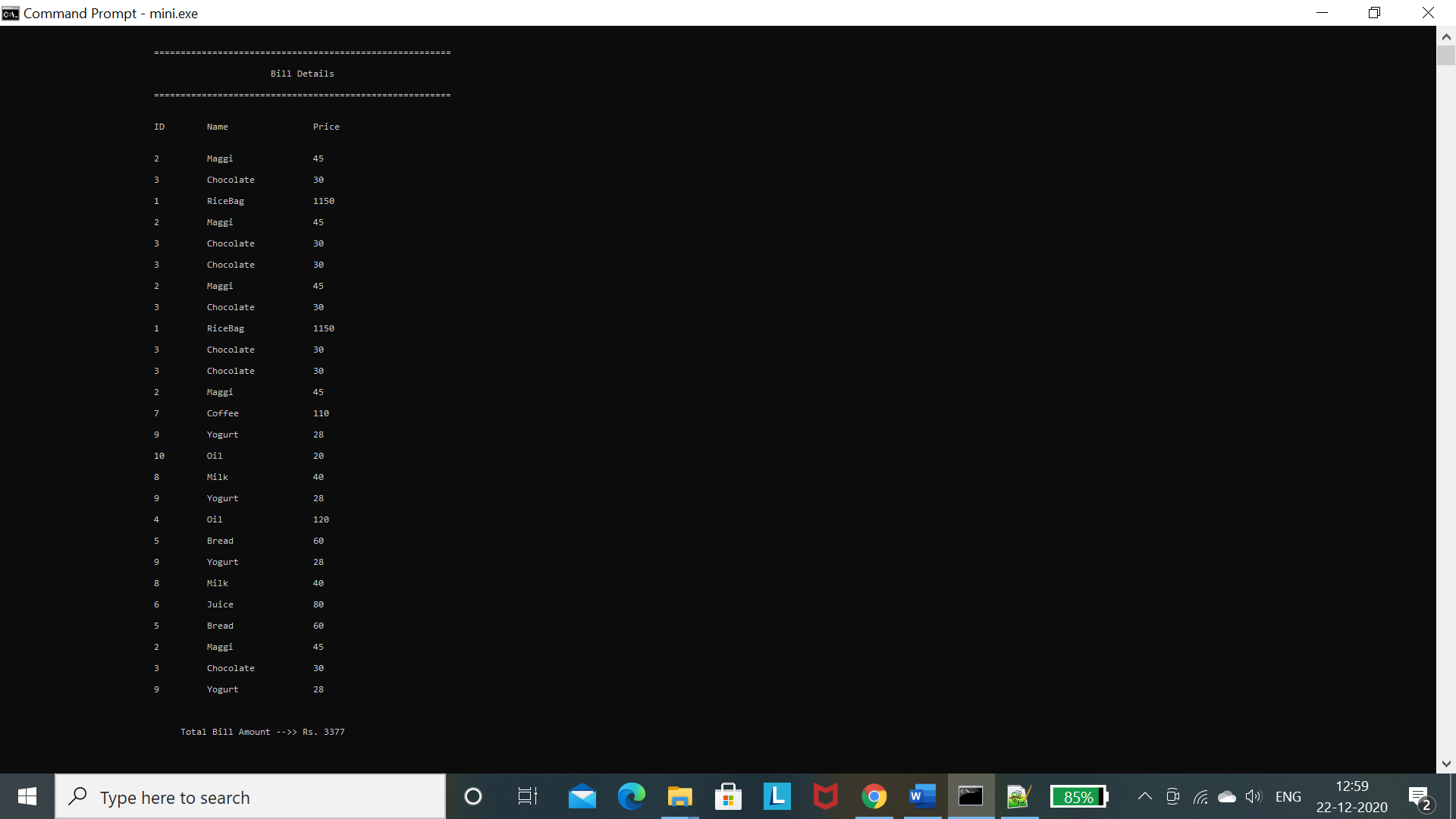
printf("Total Bill Amount -->> Rs. %d ",total);

fclose(fp);

total=0;

getch();

}



**11. Purchase Products**

**Description:** Allows the customer to view all the products along with their available quantity and price to purchase.

void purchaseProduct(char user[])

{

char ch1;

int id,lastId;

clrscr();

while(1)

{

displayAllProducts();

fflush(stdin);

printf("\n\nDo you want to purchase [y/n] : ");

scanf("%c",&ch1);

if(ch1=='y')

{

FILE \*fp,\*fp1;

struct Bill t1;

struct Product t2;

fp=fopen(fbill,"ab");

printf("\n\nEnter the Product ID to Purchase : ");

scanf("%d",&id);

fp1=fopen("id.txt","r");

fscanf(fp1,"%d",&lastId);

fclose(fp1);

if((lastId<id) || (id<=0))

{

printf("\n\n-> There is no product with that ID !");

getch();

}

else

{

t2=findProduct(id);

if(t2.qty!=0)

{

t1.pid=t2.id;

strcpy(t1.pname,t2.name);

strcpy(t1.user,user);

t1.pprice=t2.price;

fwrite(&t1,sizeof(t1),1,fp);

}

else

{

printf("\n\n-> The product with this ID is OUT OF STOCK !");

getch();

}

}

fclose(fp);

}

fflush(stdin);

clrscr();

if(ch1=='n')

{

break;

}

if(ch1!='y' && ch1!='n')

{

printf("\n\n-->>Enter 'y' or 'n' !! \n\n\t\t Press any key to continue..");

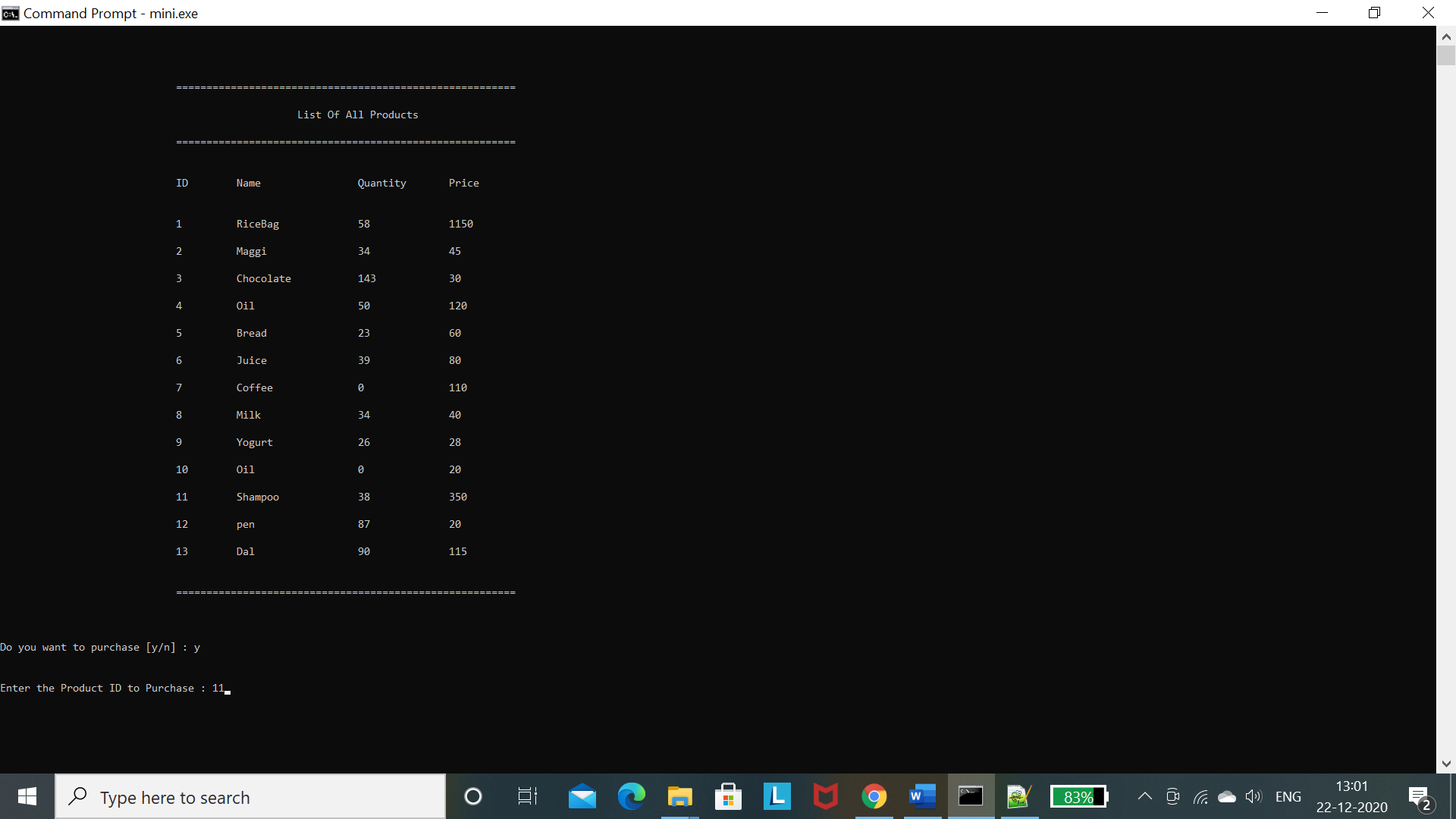
getch();

}

}

generateBill(user);

}



**12. Generate Bill**

**Description:** Asks the customer if they want to generate the final bill, if yes it will generate the final bill, else the customer can continue shopping.

void generateBill(char user[])

{

FILE \*fp,\*fp1,\*fp2;

struct Bill t;

int id,found=0;

char ch1;

int brel=0;

char billName[20];

clrscr();

fp=fopen(fbill,"rb");

gotoxy(30,5);

printf("========================================================");

gotoxy(30,7);

printf("\t\t\t Bill Details");

gotoxy(30,9);

printf("========================================================");

gotoxy(30,12);

printf("ID");

gotoxy(40,12);

printf("Name");

gotoxy(60,12);

printf("Price");

int a=15;

while(1)

{

fread(&t,sizeof(t),1,fp);

if(feof(fp))

{

break;

}

gotoxy(30,a);

printf("%d\t",t.pid);

gotoxy(40,a);

printf("%s\t",t.pname);

gotoxy(60,a);

printf("%d\t\n",t.pprice);

total=total+t.pprice;

a=a+2;

}

gotoxy(35,a+2);

printf("Total Bill Amount -->> Rs. %d ",total);

fclose(fp);

if(total!=0)

{

fflush(stdin);

printf("\n\n\n\nDo you want to generate the Final Bill [y/n] : ");

scanf("%c",&ch1);

if(ch1=='y')

{

brel=billFileNo();

sprintf(billName,"%s%d","",brel);

strcat(billName,".dat");

fp=fopen(fbill,"rb");

fp1=fopen(billName,"wb");

fp2=fopen(fAll,"ab");

while(1)

{

fread(&t,sizeof(t),1,fp);

if(feof(fp))

{

break;

}

fwrite(&t,sizeof(t),1,fp1);

fwrite(&t,sizeof(t),1,fp2);

}

fclose(fp);

fclose(fp1);

fclose(fp2);

fp=fopen(fbill,"wb");

fclose(fp);

total=0;

}

else

{

total=0;

purchaseProduct(user);

}

}

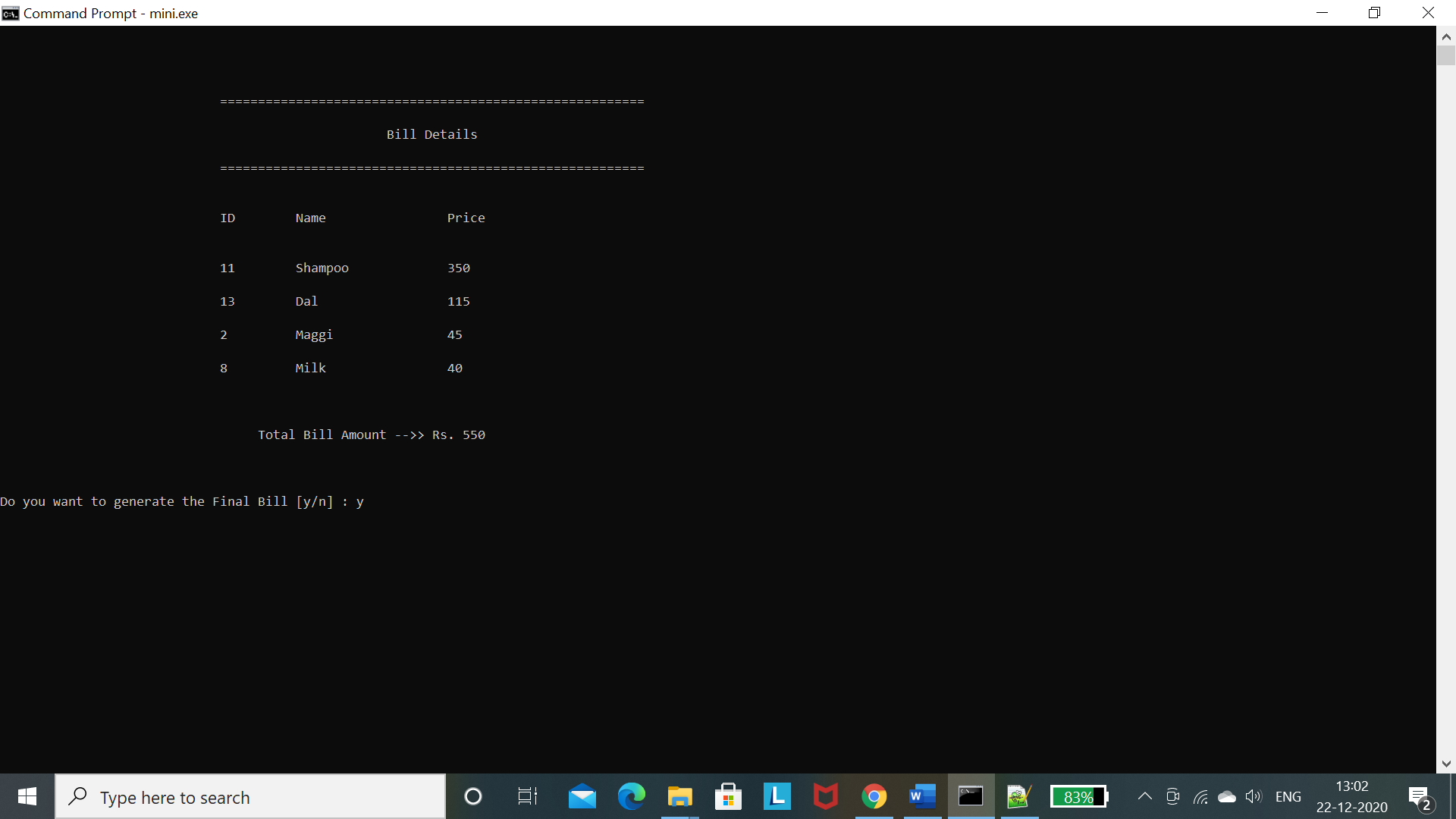
else

{

printf("\n\n\n\n\t\t\t\t You have not purchased any product !! \n");

}

}



**13. Previous Bills**

**Description:** Allows customers to view all their previous bills. (Same as specific bills module using username)

**ii. Github links**

<https://github.com/Abignya/mini-project>

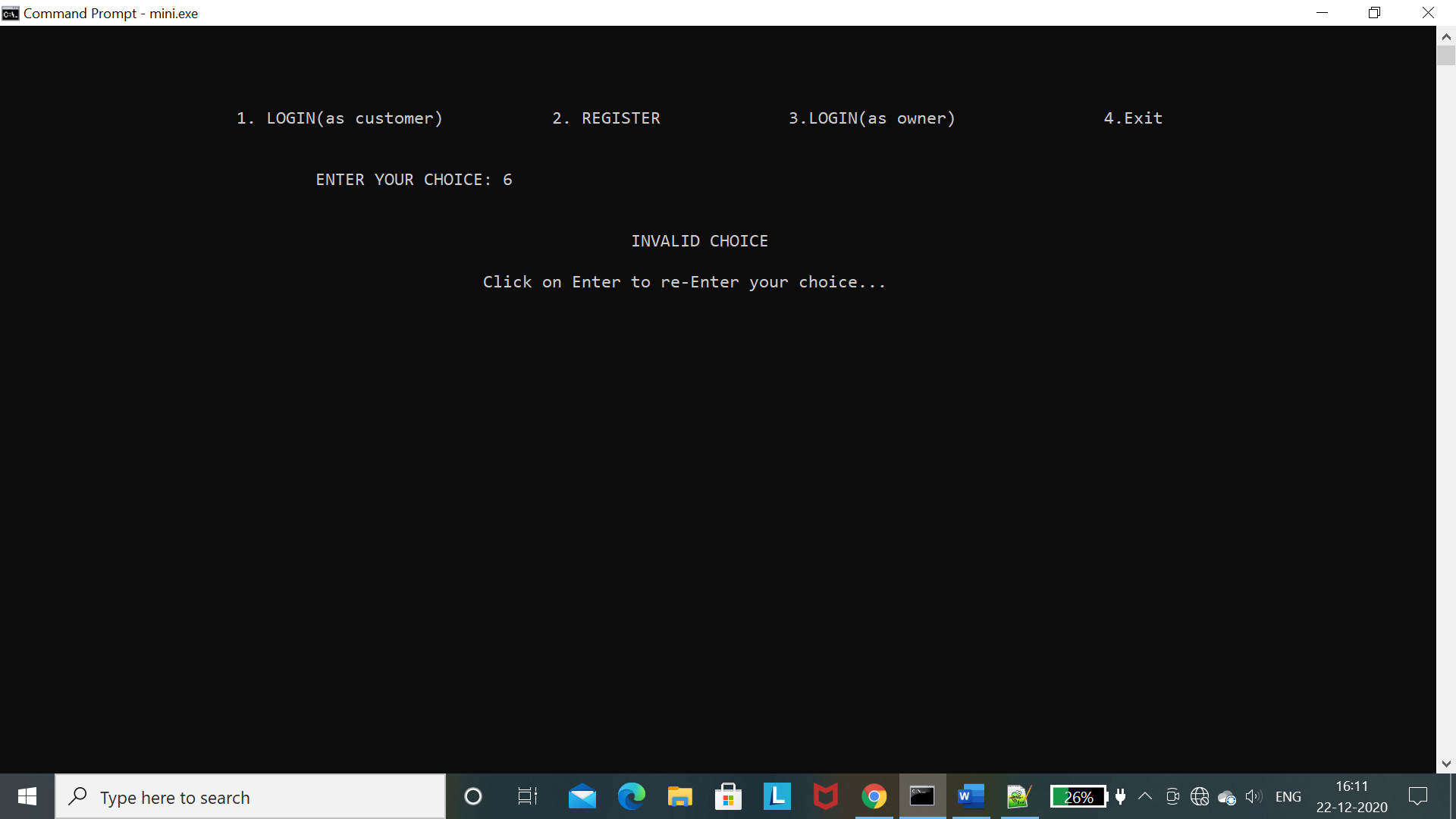
<https://github.com/freyarachel/mini-project>

**c. Testing**

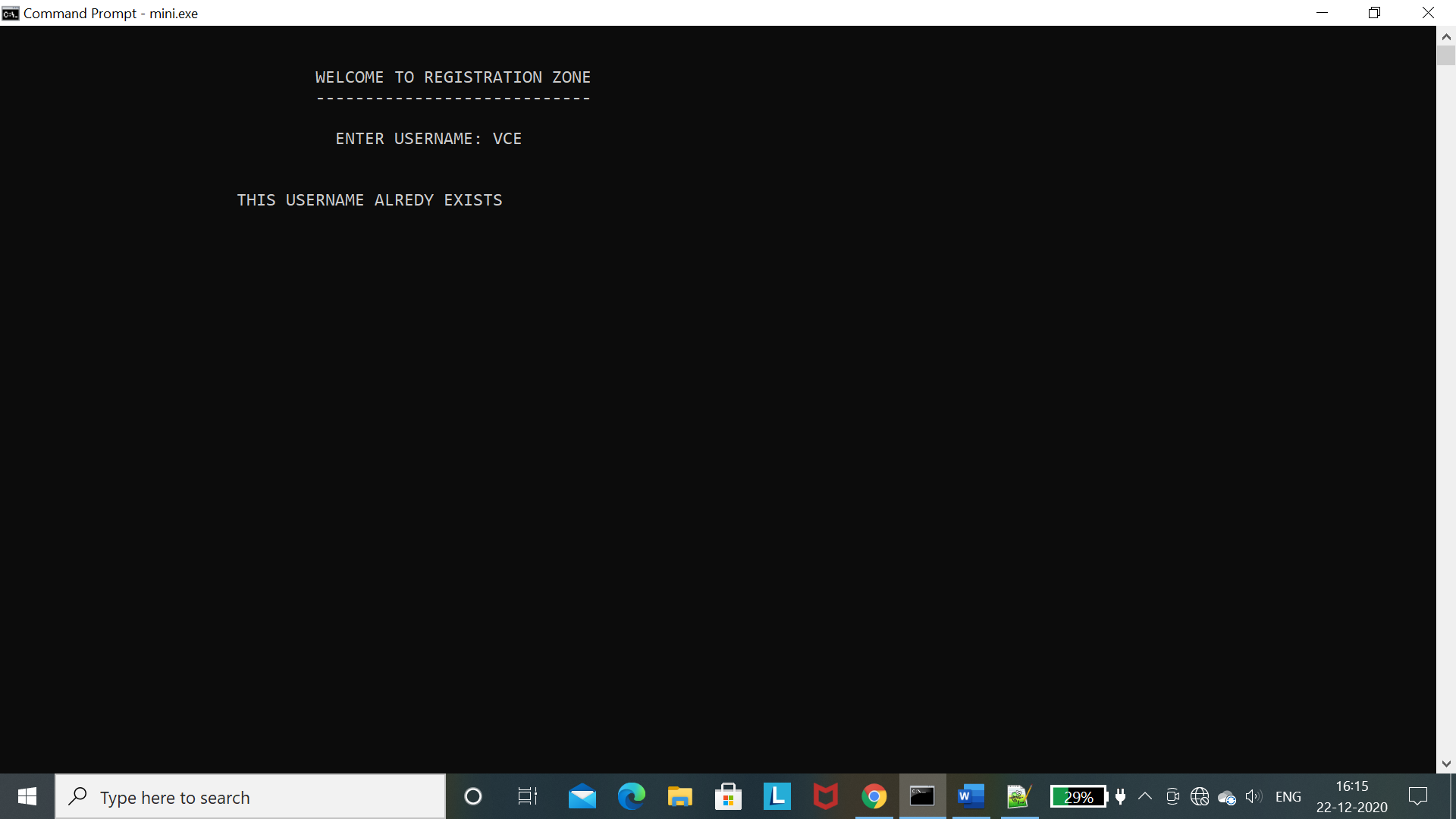
The code written for building Grocery management system and connecting with database ensures that the values entered by the user are correct. It prompts an error message if the values are not entered correctly.

**Example**

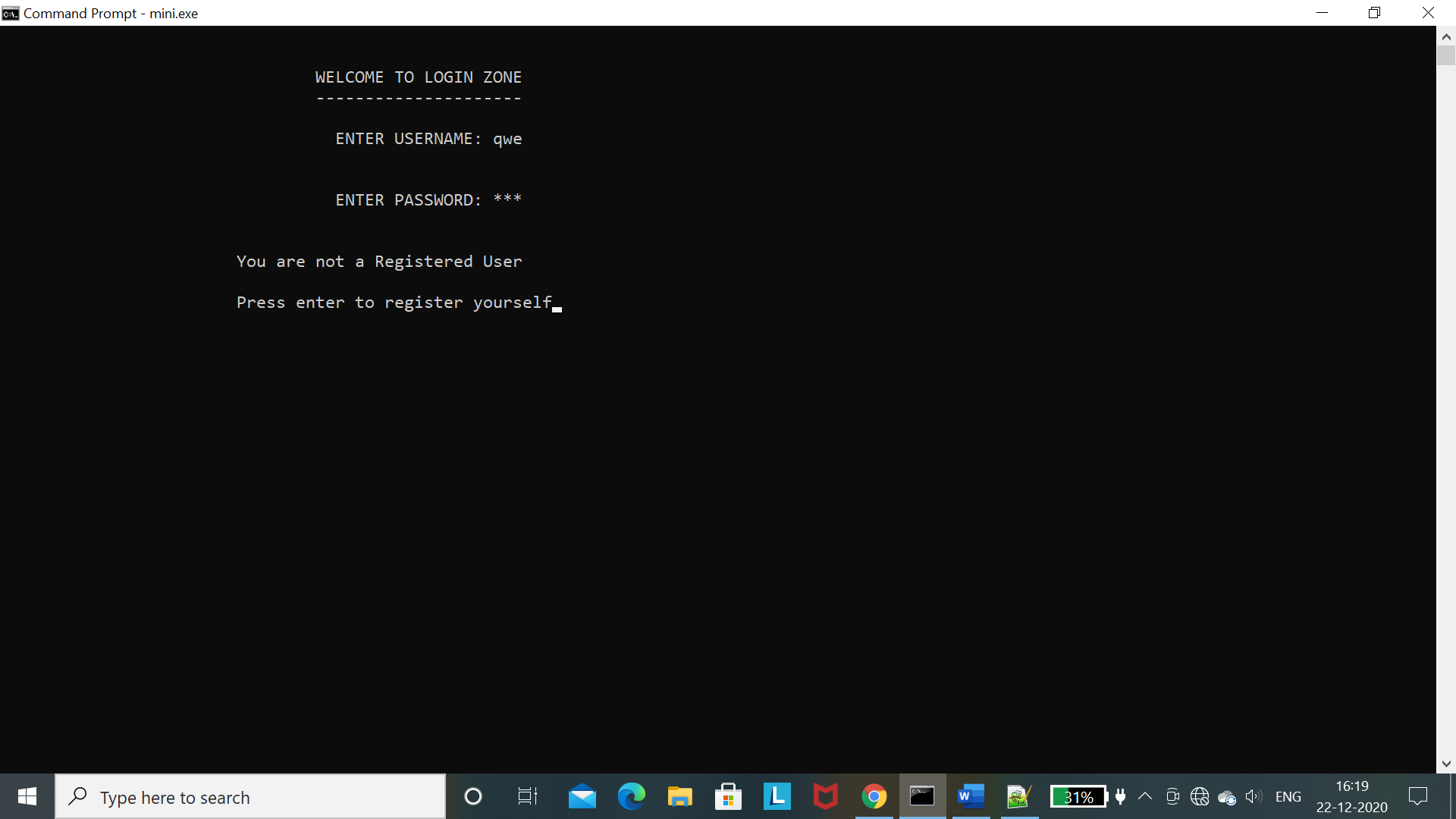
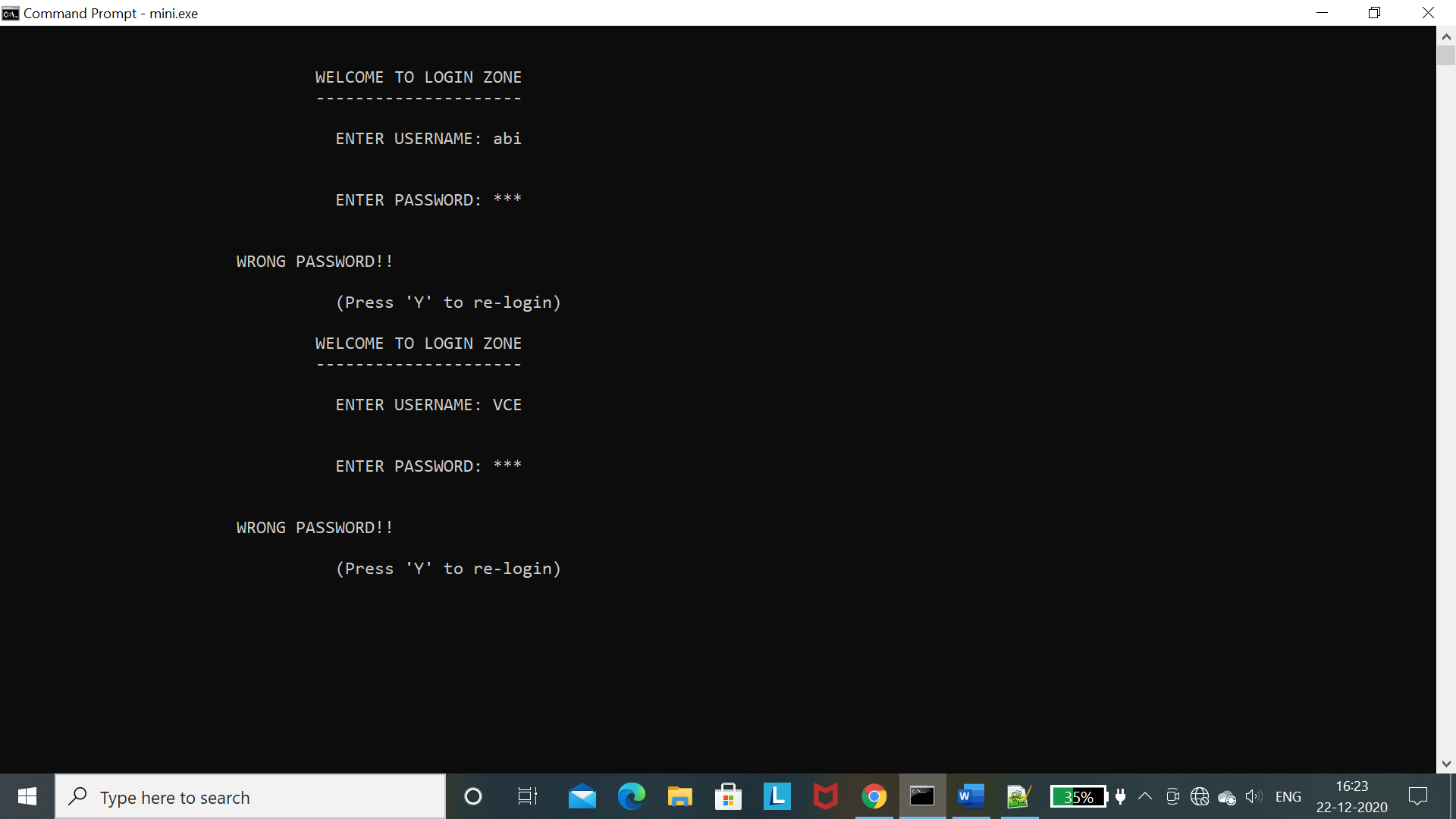
* Every time a menu appears and user gives wrong input



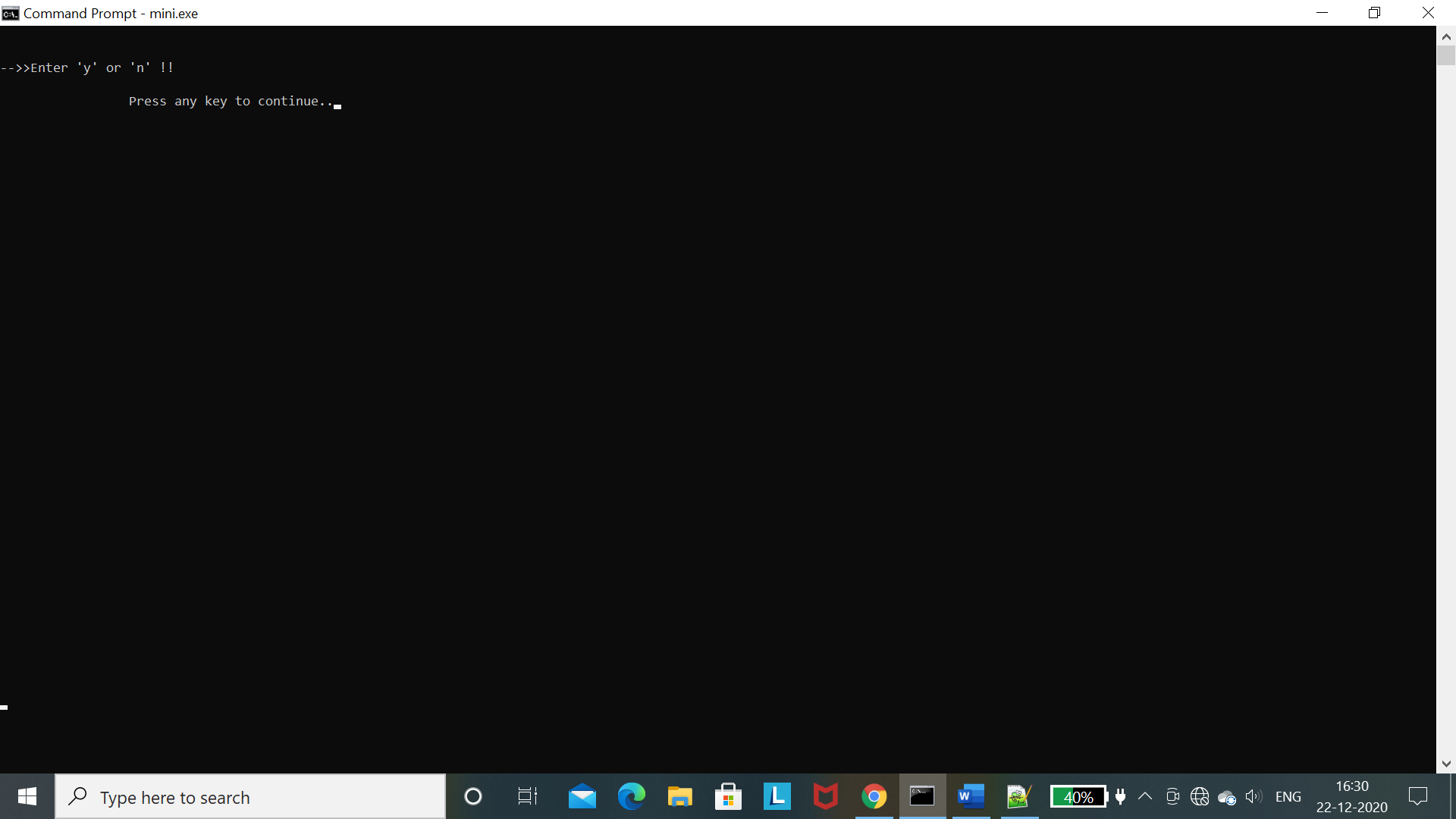
* When a new customer registers with already existing username

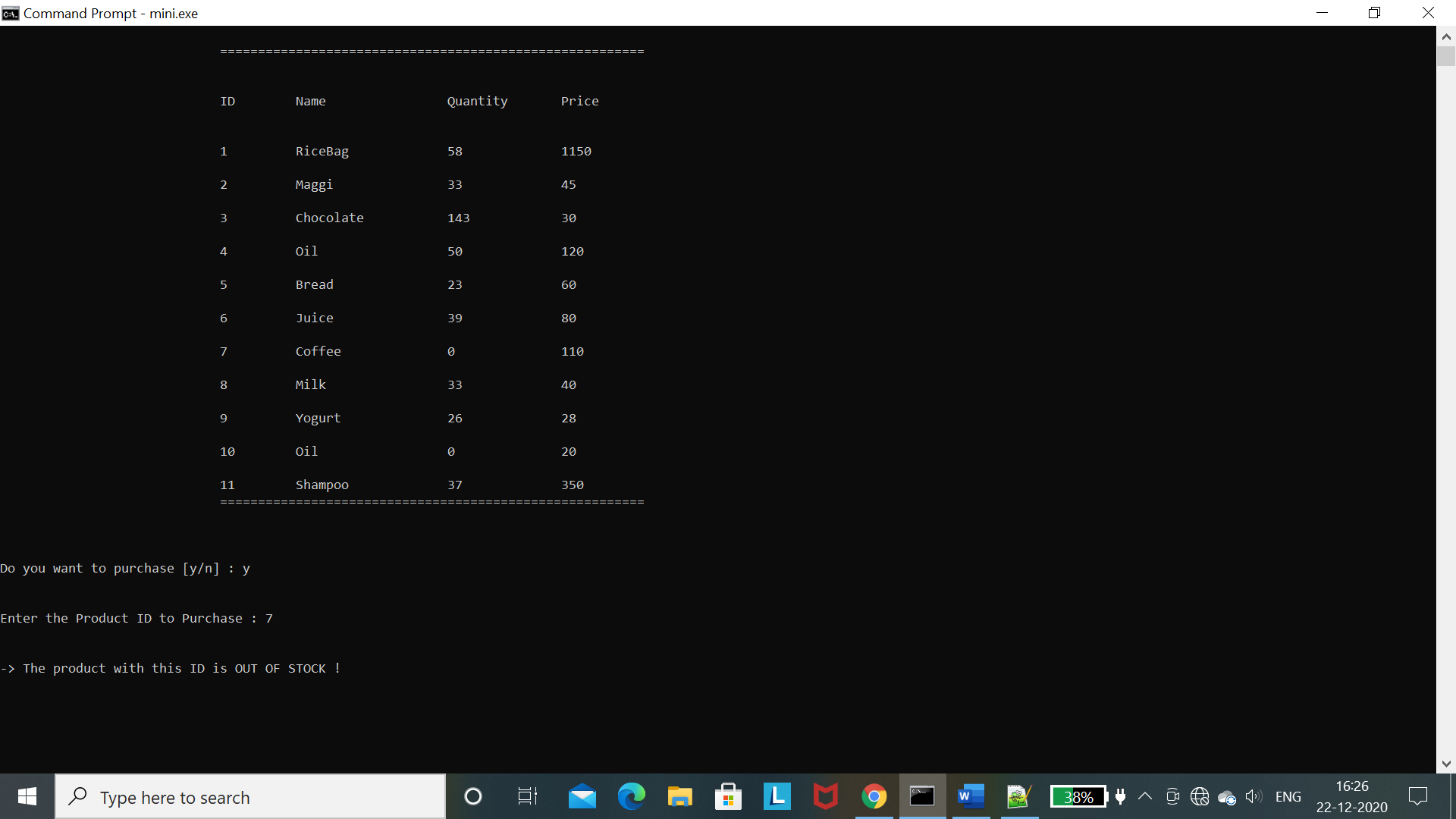


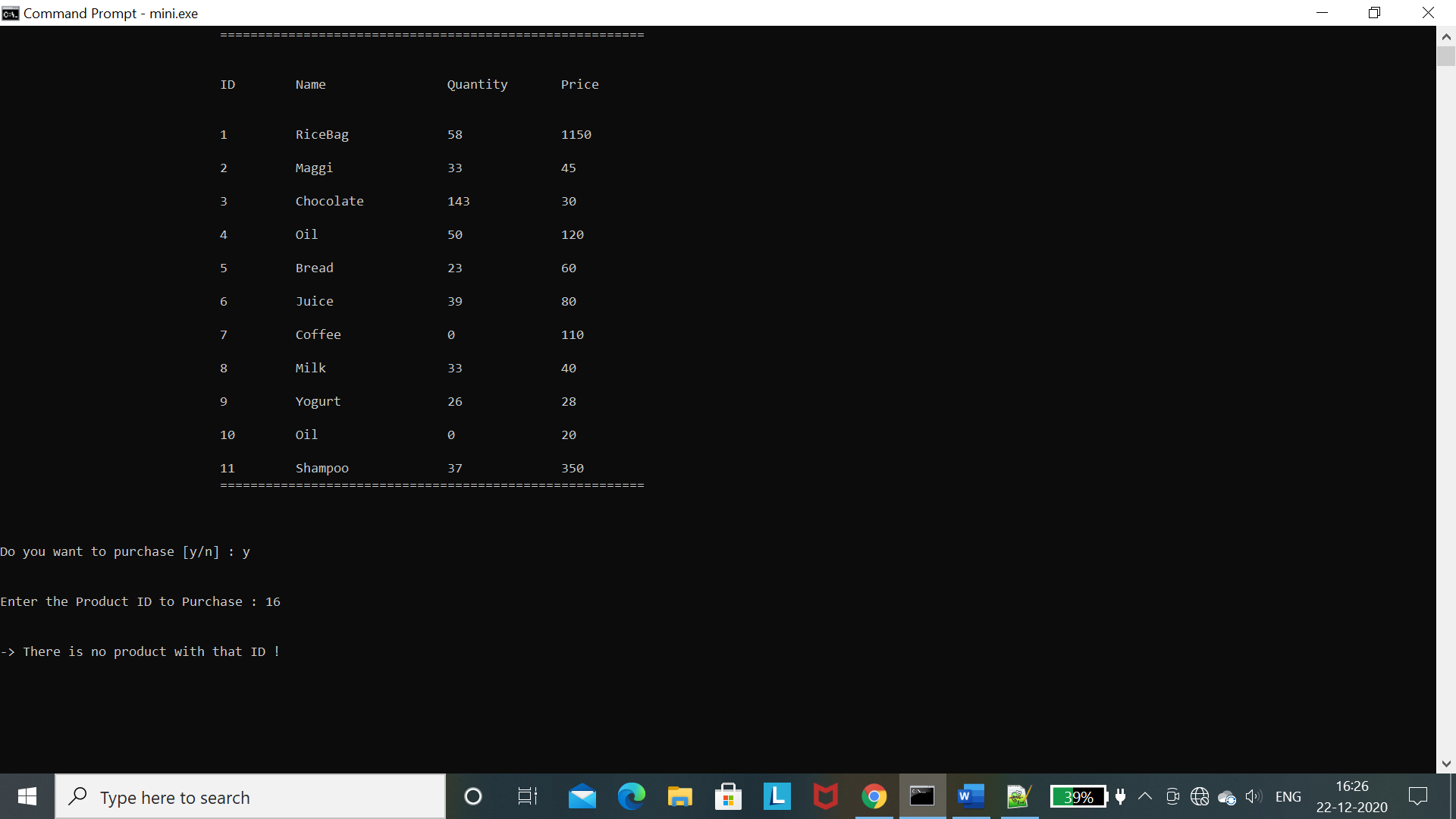
* When customer enters wrong username or password

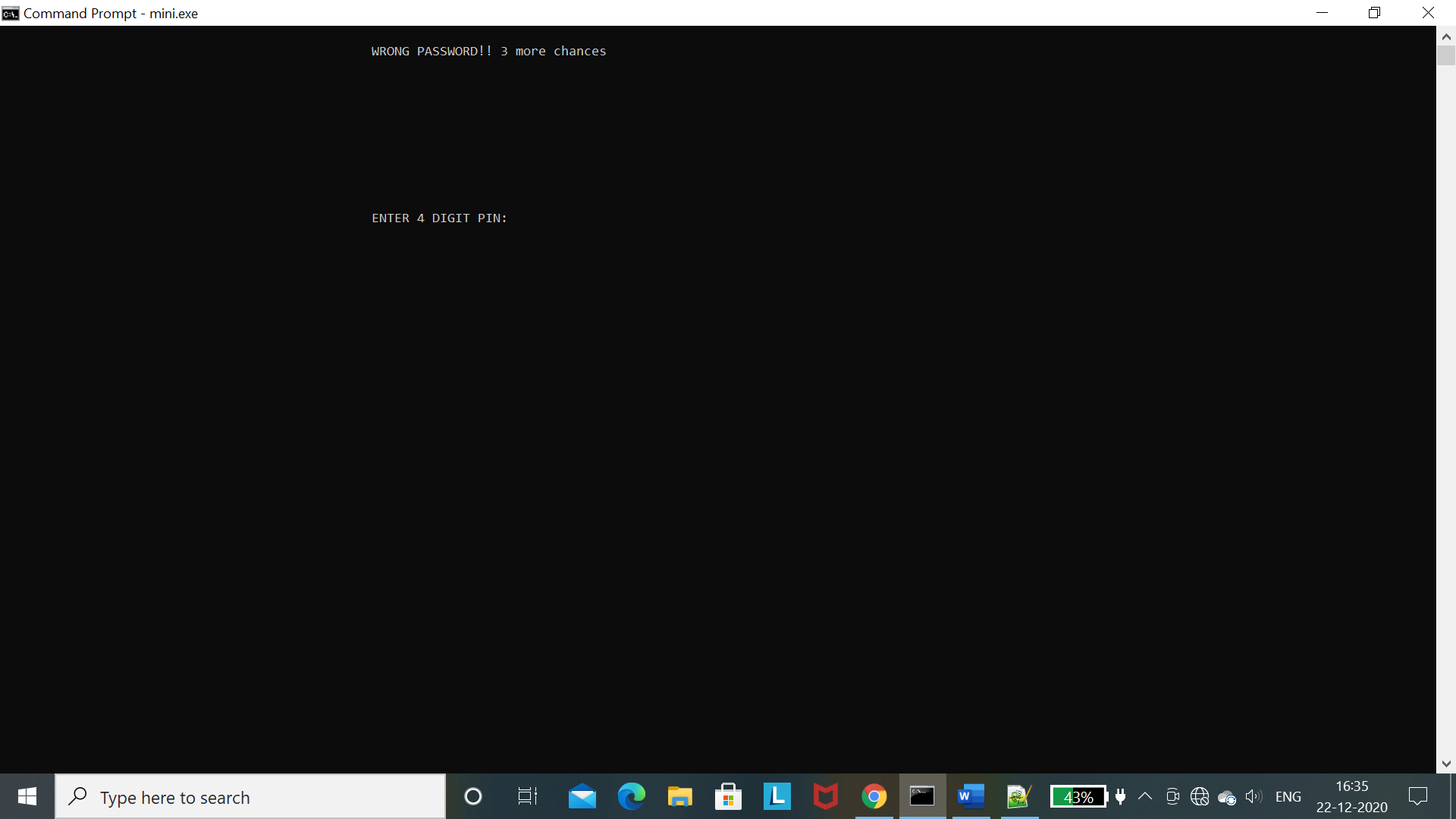
* While purchasing if we enter invalid product ID or enter a product ID with 0 quantity



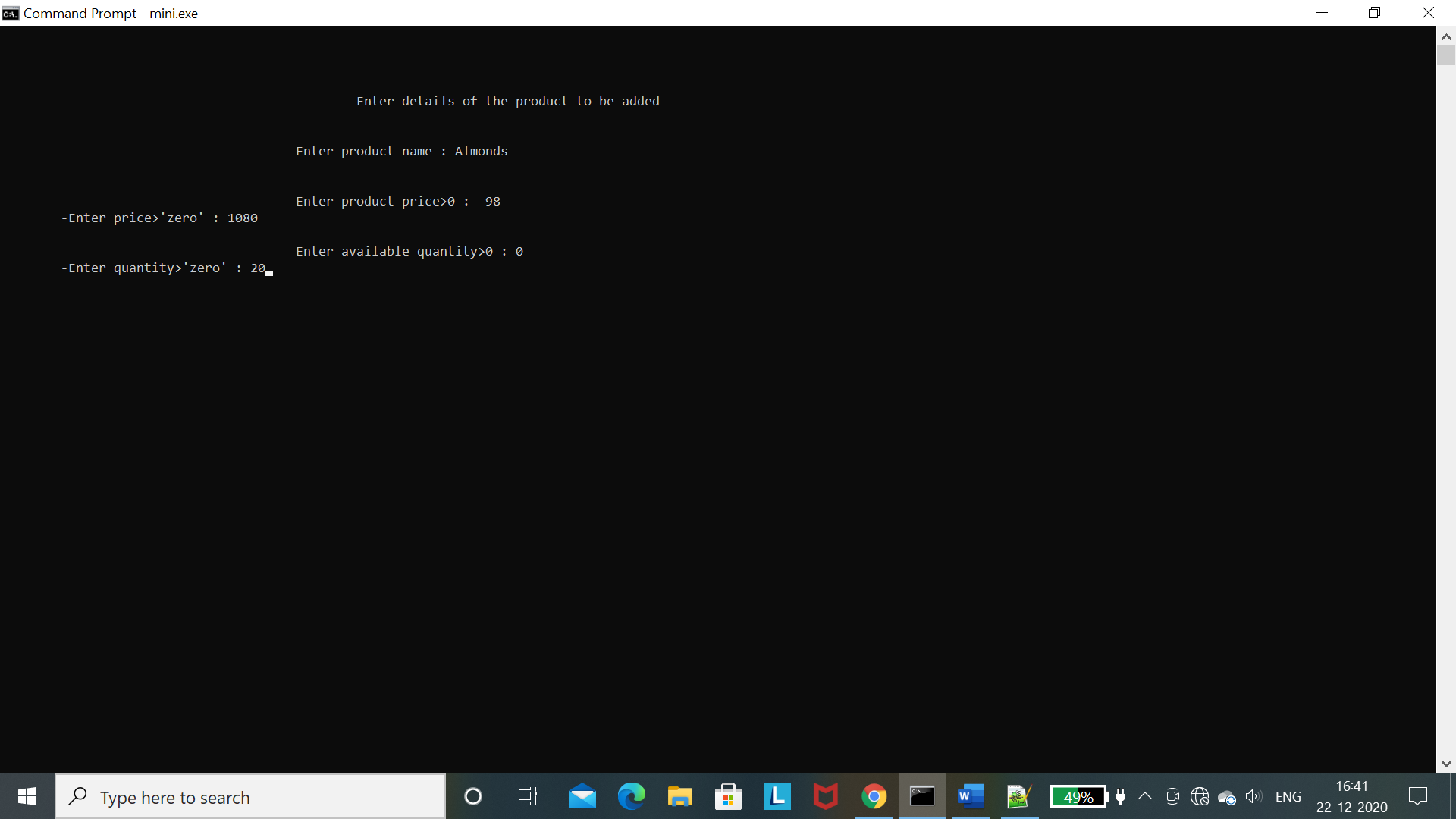




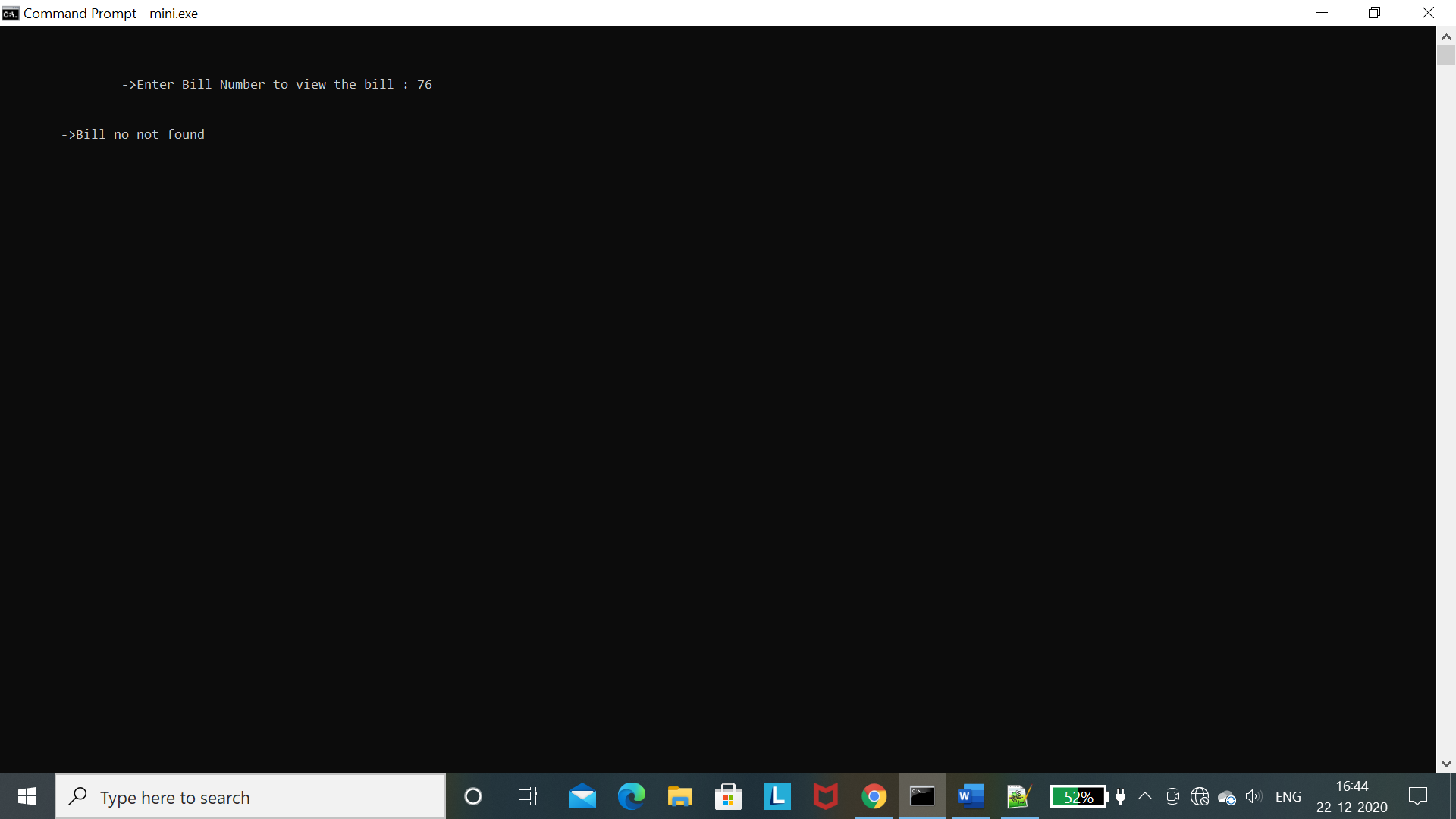
* When the owner enters wrong password



* While adding a product



* When owner enters wrong Bill number



M. Abignya Reddy

1602-19-737-001

G.R. Freya Rachel

1602-19-737-032

**Additional Learning**

This project is a great experience to learn many new things. It was very helpful

in improving our coding skills and concepts related to C. Teamwork is another

important skill in which we improved ourselves from sharing our ideas to presenting the

final project. It helped us in organizing our thoughts, planning our work, time

management, communication skills and learning together. Working as a team also

helped in completing the work in less time and in more efficient way and sharing the

workload eases the burn out.

Self-learning is another important aspect which we learnt while working for

this project. We learnt many new concepts and different ways of approach to the

problem, fixing the errors, debugging etc. We also had a thorough revision of all the concepts and made our foundation much stronger. It also helped us to be more creative in various steps of its development. It is an opportunity to enhance our skills.

M. Abignya Reddy

1602-19-737-001

G.R. Freya Rachel

1602-19-737-032

**Discussion and Future Work**

In the proposed ‘Grocery Management’ we want to enhance this system by including other functionalities like exchange, repetition of the same order every month (Calendar System), discounts etc. We also want to include data structures which will make the application faster and organized. Developing this to online web application also makes it accessible to people easily. Providing SMS and email services to customers visiting the store, including online bill payment. And developing the code with Java, HTML using graphics makes the application more user friendly.

M. Abignya Reddy

1602-19-737-001

G.R. Freya Rachel

1602-19-737-032

**References**

We have referred the following sites:

<https://www.geeksforgeeks.org/c-programming-language/>

<https://www.tutorialspoint.com/cprogramming/index.htm>

M. Abignya Reddy

1602-19-737-001

G.R. Freya Rachel

1602-19-737-032