SUPERMARKET MANAGEMENT SYSTEM

- *Working description of project*
- *Header files and their purpose*
- *Files generated*
- *Coding & outputs*

INTRODUCTION

This project makes the task of ordering products from a supermarket easier. The task of maintaining individual order details and amounts for administration purposes are also taken care of by the program.

It enables the administrator to enter a new product, modify and delete product details, view all order details, view various products in the inventory. Its user friendly approach and easy to comprehend tasks makes it a useful tool both for the customer and the administration. Ordering items from existing inventory with the product number, name and price enables customers to place their orders quickly and efficiently. The total billing amount after discounts is also displayed simultaneously.

The concern for security of administration has been taken into account by keeping a password controlled access to administrator menu.

HEADER FILES USED AND THEIR PURPOSE

1. **IOSTREAM.H** – *for cin and cout operations*
2. **PROCESS.H** – *for exit() function*
3. **CONIO.H** – *for clrscr() and getch() functions*
4. **STDIO.H** – *for standard I/O operations*
5. **STRING.H** – *for string handling*
6. **FSTREAM.H** – *for file handling(reading and writing in file)*

FILES GENERATED

PROGRAM FILE

CS1.CPP

OBJECT FILE

CS1.OBJ

EXECUTION FILE

CS1.EXE

FILE FOR STORING PRODUCT DETAILS

SHOP.TXT

FILE FOR ORDER DETAILS

ORDER.TXT

TEMPORARY FILE FOR RECORD MODIFICATION

TEMP.TXT

```
//*************************************************
//      PROJECT : Supermarket Management System
//*************************************************

//*************************************************************
// HEADER FILE USED IN PROJECT
//*************************************************************
#include<iostream.h>
#include<conio.h>
#include<stdio.h>
#include<process.h>
#include<fstream.h>
#include<string.h>

//*************************************************************
// CLASS USED IN PROJECT
//*************************************************************
class product
{
        int prono;
        char name[50];
        float cost, qty, disc;
        public:
        void create_product()
        {
                cout << "\nenter name of product: ";
                gets(name);
                cout << "\n\nenter unique product number: ";
                cin>>prono;
                cout << "\n enter the cost of product: ";
                cin >> cost;
                cout << "\nenter the discount available on product: ";
                cin >> disc;
        }
        void show_product()
        {
                cout << "\nThe Product No. of The Product : " << prono;
                cout << "\nThe Name of The Product : ";
                puts(name);
                cout << "\nThe Price of The Product : " << cost;
                cout << "\nDiscount : " << disc;
        }
                int getprono()
        {
                return prono;
        }
                float getcost()
        {
                return cost;
        }
                char * retname()
        {
                return name;
        }
```

```
		int getdisc()
	{
		return disc;
	}
	}; //class ends here
class order
{
	public:
	int ono,pno;
	int qty;
	int amt,damt;
	void showorder()
	{
		cout<<"\nOrder No:"<<ono;
		cout << "\nOrder Product No. : " << pno;
		cout << "\nOrder Quantity : "<<qty;
		cout<<"\nOrder Amount:"<<amt;
		cout << "\nDiscounted amount : " << damt;
	}
}o;
//*************************************************************
// global declaration for stream object, object
//*************************************************************
fstream fp;
product prod;
int ocnt=0;
void mainmenu();

//*************************************************************
// function for generating bill for Products
//*************************************************************
int order_count()
{
	ifstream f;
	f.open("d:\\shivam\\order.txt", ios:: in|ios::binary );

	int cnt=0;

	if(f)
	{
		f.read((char * ) & o, sizeof(o));
		while(!f.eof())
		{
			cnt++;
			f.read((char * ) & o, sizeof(o));

		}
	}

	cout<<cnt;
	fp.close();
	return cnt;
}


//*************************************************************
```

```
//function for placing orders
//***************************************************************
void place_order()
{
        int pno,qty,amt,damt;
        fp.open("d:\\shivam\\Shop.txt", ios:: in|ios::binary );

        cout << "\n\n\t\tProduct MENU\n\n";
        cout << "\n";
        cout << "P.NO.\t\tNAME\t\tPRICE\n";
        cout << "\n";

        while (fp.read((char * ) & prod, sizeof(prod)))
        {
                cout << prod.getprono() << "\t\t" << prod.getcost() << "\t\t" <<
                prod.getdisc() << endl;
        }
        fp.close();

        cout << "\n PLACE YOUR ORDER";
        cout<<endl<<endl;
        cout<<"enter product no to order:";
        cin>> pno;
        cout<<"enter quantity:";
        cin>>qty;

        ofstream fout;
        fout.open("d:\\shivam\\order.txt", ios:: app|ios::binary );

        order o;

        fp.open("d:\\shivam\\Shop.txt", ios:: in|ios::binary );
        fp.read((char * ) & prod, sizeof(product));

        while (!fp.eof())
        {
                        if (prod.getprono() == pno)
                        {
                                o.ono=order_count()+1;
                                o.pno=pno;
                                o.qty=qty;
                                o.amt = prod.getcost() * qty;
                                o.damt = o.amt - (o.amt * prod.getdisc() / 100);
                                fout.write((char * ) & o, sizeof(order));
                                break;
                        }
                        fp.read((char * ) & prod, sizeof(product));
                }
        cout<<endl<<"the total billing amount is:"<<o.damt<<endl;
        cout << "\n\nThanks for placing order";
        fout.close();
        fp.close();
        getch();
        clrscr();
}
```

```
//***************************************************************
// function to write in file
//***************************************************************

void write_product()
{
	fp.open("d:\\shivam\\Shop.txt", ios::out |ios::binary| ios::app);

	prod.create_product();

	fp.write((char * ) & prod, sizeof(product));

	fp.close();

	cout << "\n\nThe Product Has Been Created ";
	getch();
}
//***************************************************************
// function to read all records from file
//***************************************************************
void display_all()
{
	clrscr();
	cout << "\n\n\n\t\tDISPLAY ALL RECORD !!!\n\n";

	fp.open("d:\\shivam\\Shop.txt", ios:: in );
	while (fp.read((char * ) & prod, sizeof(product)))
	{
		prod.show_product();
		cout << "\n\n====================================\n";
		getch();
	}
	fp.close();
	getch();
}
//***************************************************************
// function to read specific record from file
//***************************************************************
void display_sp(int n)
{
	int flag = 0;
	fp.open("d:\\shivam\\Shop.txt", ios:: in|ios::binary );

	while (fp.read((char * ) & prod, sizeof(product)))
	{
		if (prod.getprono() == n)
		{
			clrscr();
			prod.show_product();
			flag = 1;
		}
	}
```

```
	fp.close();

	if (flag == 0)
		cout << "\n\no such record entered exists in inventory.";
	getch();
}




//***************************************************************
// function to modify record of file
//***************************************************************
void modify_product()
{
	int no, found = 0;
	clrscr();
	cout << "\n\n For modification enter product no of record to be updated";
	cin >> no;

	fp.open("d:\\shivam\\Shop.txt", ios:: in | ios::out|ios::binary);

	while (fp.read((char * ) & prod, sizeof(product)) && found == 0)
	{
		if (prod.getprono() == no)
		{
			prod.show_product();
			cout << "\nenter new details of the product" << endl;
			prod.create_product();
			int pos = -1 * sizeof(prod);
			fp.seekp(pos, ios::cur);
			fp.write((char * ) & prod, sizeof(product));
			cout << "\n\n\record has been updated in the system";
			found = 1;
		}
	}

	fp.close();
	if (found == 0)
		cout << "\n\n Record Not Found ";
	getch();
}
```

```
//***************************************************************
// function to delete record of file
//***************************************************************
void delete_product()
{
	int no;
	clrscr();
	cout << "\n\n\n\tDelete Record";
	cout << "\n\nPlease Enter The product no. of The Product You Want To Delete";
	cin >> no;

	fp.open("d:\\shivam\\Shop.txt", ios:: in |ios::binary);
	fstream fp2;
	fp2.open("d:\\shivam\\Temp.txt", ios::out|ios::binary);
	fp.seekg(0, ios::beg);

	while (fp.read((char * ) & prod, sizeof(product)))
	{
		if (prod.getprono() != no)
		{
			fp2.write((char * ) & prod, sizeof(product));
		}
	}

	fp2.close();
	fp.close();
	remove("d:\\shivam\\Shop.txt");
	rename("d:\\shivam\\Temp.txt", "d:\\shivam\\Shop.dat");
	cout << "\n\n\tRecord Deleted ..";
	getch();
}



//***************************************************************
// function to display all order details
//***************************************************************

void display_orders()
{
	//clrscr();
	int eof;
	fp.open("d:\\shivam\\order.txt", ios::in|ios::binary);

	if (!fp)
	{
		cout << "ERROR!!! FILE COULD NOT BE OPEN\n\n\n Go To Admin Menu to
			create file";

		cout << "\n\n\n Program is closing ....";
		getch();
		exit(0);
	}
	else
	{
```

```
                while (fp.read((char * ) & o, sizeof(o)))
                {
                                o.showorder();
                                getch();
                }
                fp.close();

        }

}
//*****************************************
//ADMINISTRATOR MENU FUNCTION
//*****************************************
void admin_menu()
{
        clrscr();
        int ch2;
        do{
        cout << "\n\n\n\tadministrator menu";
        cout << "\n\n\t1.add product details";
        cout << "\n\n\t2.show all existing products";
        cout << "\n\n\t3.enter product no to get its data displayed as a query ";
        cout << "\n\n\t4.modify product details";
        cout << "\n\n\t5.delete product";
        cout << "\n\n\t6.view all order details";
        cout << "\n\n\t7.back to main menu";
        cout << "\n\n\tselect your task from above options and enter the task no (1-7) ";
        cin>>ch2;
        switch (ch2)
        {
        case 1:
                write_product();
                break;
        case 2:
                display_all();
                break;
        case 3:
                int num;
                cout << "\n\n\tPlease Enter The Product No. ";
                cin >> num;
                display_sp(num);
                break;
        case 4:
                modify_product();
                break;
        case 5:
                delete_product();
                break;
        case 6:
                display_orders();
                break;
        case 7:
                mainmenu();
                break;
        default:
```

```
                cout << "invalid option";

  }
  }
        while(ch2!=7);
}
//*************************************************************
//THE MAIN FUNCTION OF PROGRAM
//*************************************************************
void main()
{
        mainmenu();
}
//*************************************************************
//FUNCTION FOR CHECKING PASSWORD TO ADMINISTRATOR MENU
//*************************************************************

int passwords()
{
        char p[10];
        int i=0;
        cout<<"\nENTER THE PASSWORD\n";

        do
        {
        p[i]=getch();
        cout<<"*";
        i++;
        }
        while(p[i-1]!='\r');
        p[i-1]='\0';
        if (strcmp(p,"abcde")==0)

                return 1;

        else
                return 0;
}

//*************************************************
//FUNCTION TO DISPLAY THE MAIN MENU OF BILLING SYSTEM
//*************************************************
void mainmenu()
{

  clrscr();
        char ch;
        do
        {
                clrscr();
                cout << "\n\n\n\tWelcome to MAIN MENU of supermarket billing system.";
                cout << "\n\n\t01.select 1 if you are CUSTOMER";
                cout << "\n\n\t02.select 2 if you are ADMINISTRATOR";
                cout << "\n\n\t03.select 3 if you want to EXIT";
                cout << "\n\n\tPlease Select Your Option (1-3) ";
                ch = getche();
```

```
                switch (ch)
                {
                case '1':
                        place_order();
                        getch();
                        break;
                case '2':
                        if (passwords()==1)
                                admin_menu();
                        else
                                cout<<"Invalid password";
                        break;
                case '3':
                        exit(0);
                default:
                        cout << "\a";
                }
        } while (ch != '3');
        getch();
        }


//***************************************************************
// END OF PROJECT

//***************************************************************
```

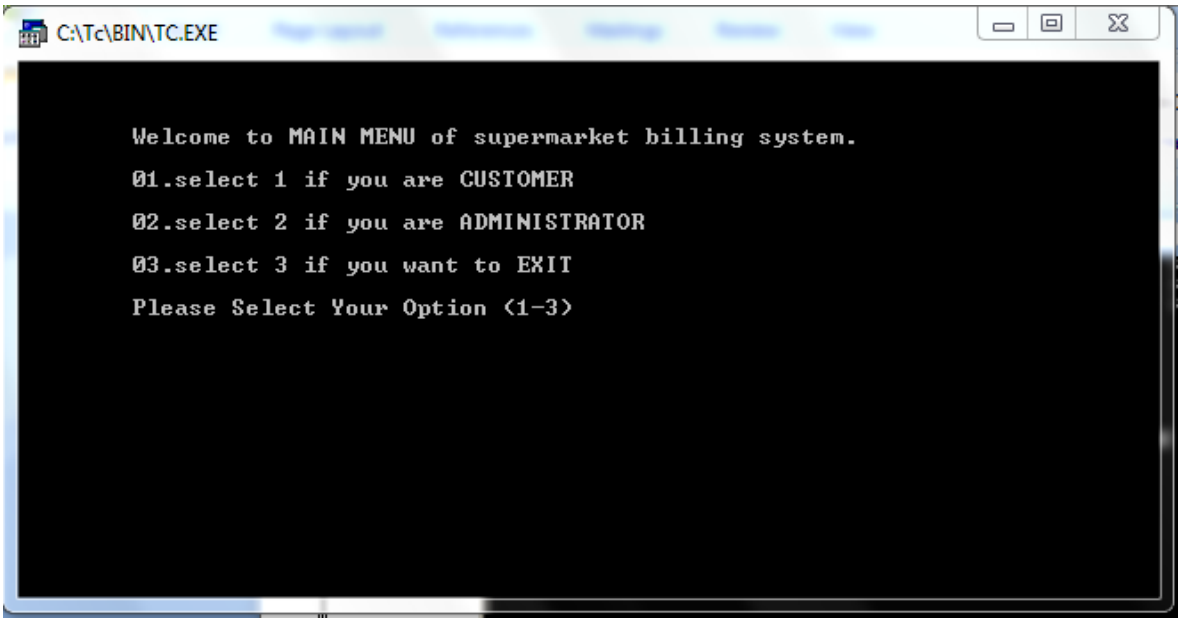
SUPERMARKET MANAGEMENT SYSTEM

Developed By:-

SHIVAM PATEL

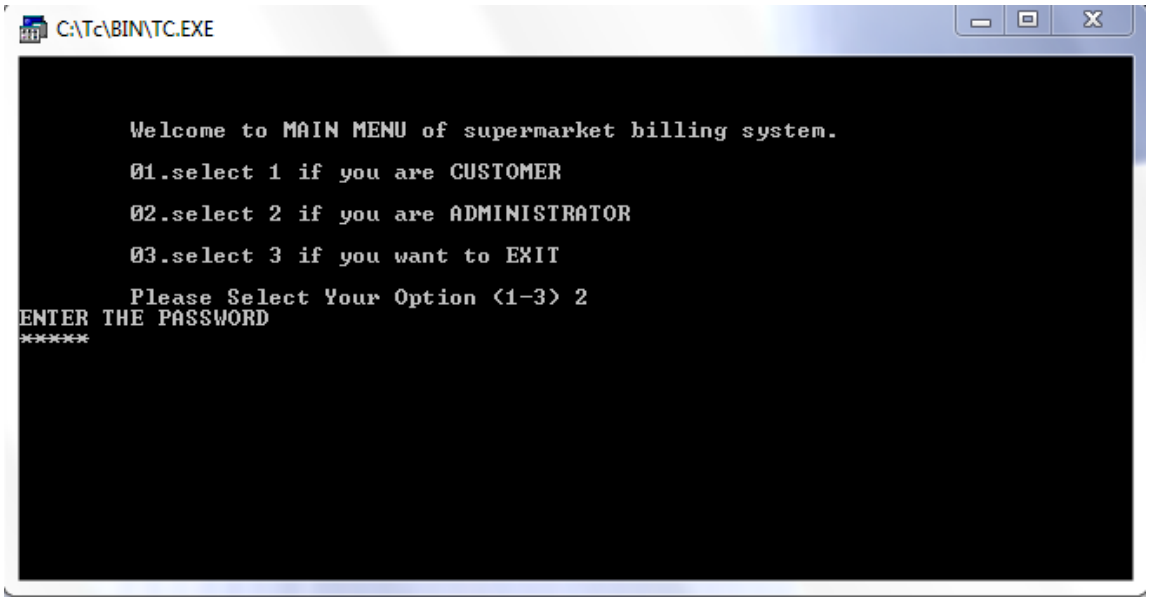
OUTPUT-SCREENS

WELCOME SCREEN

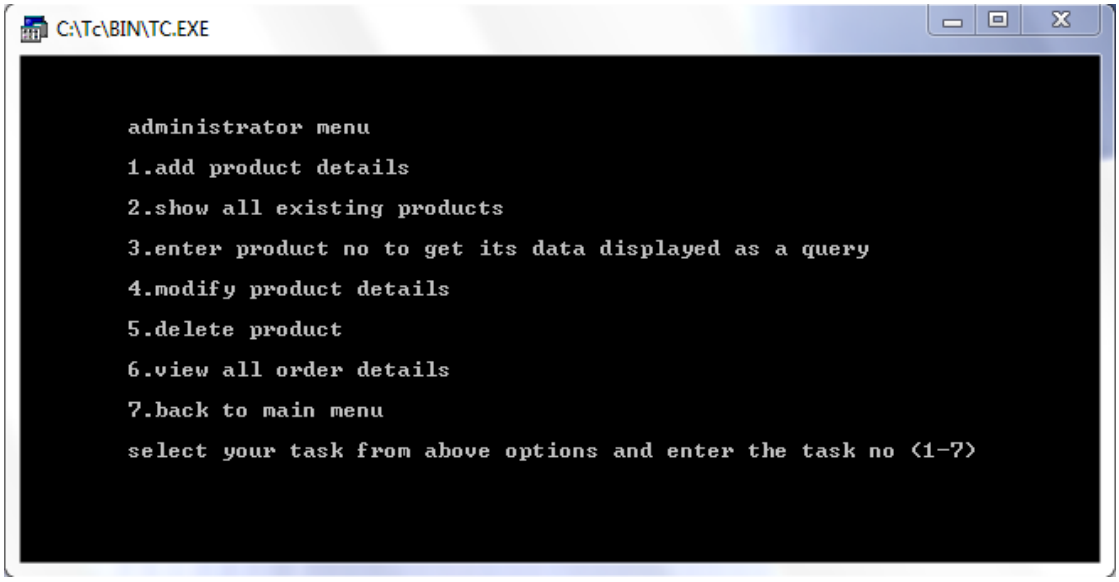


Entering password to

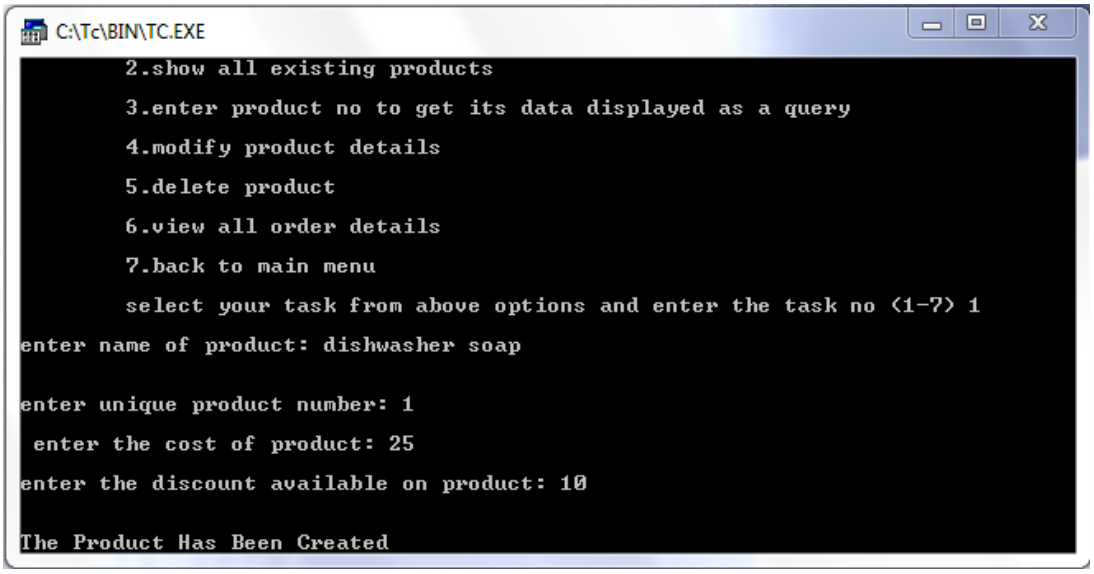
administrator menu



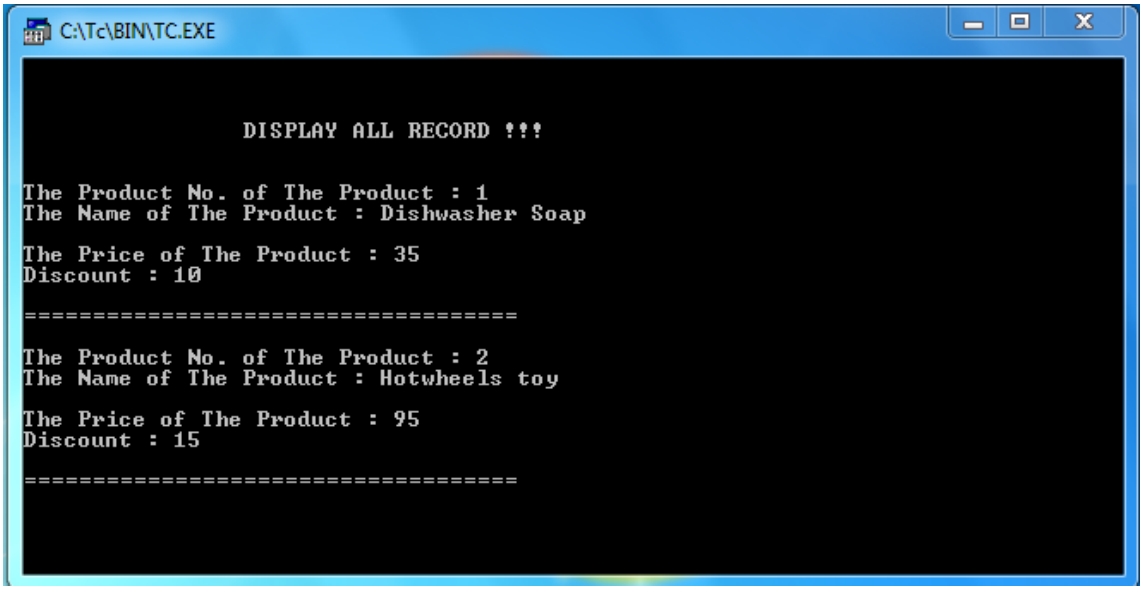
Administrator menu



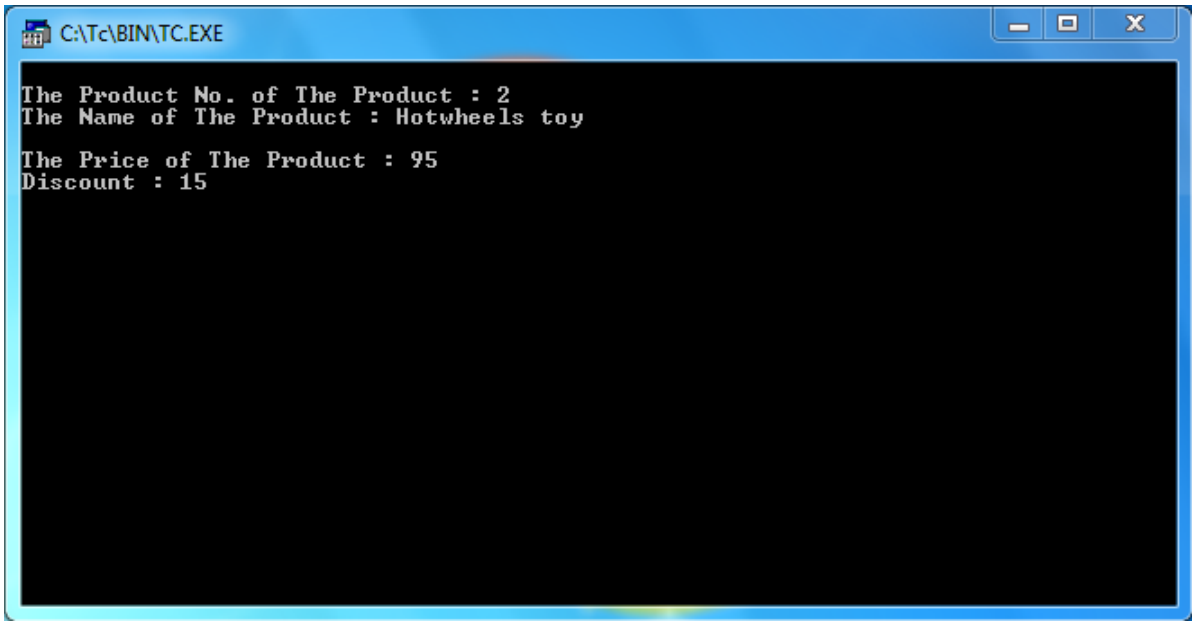
Adding product details



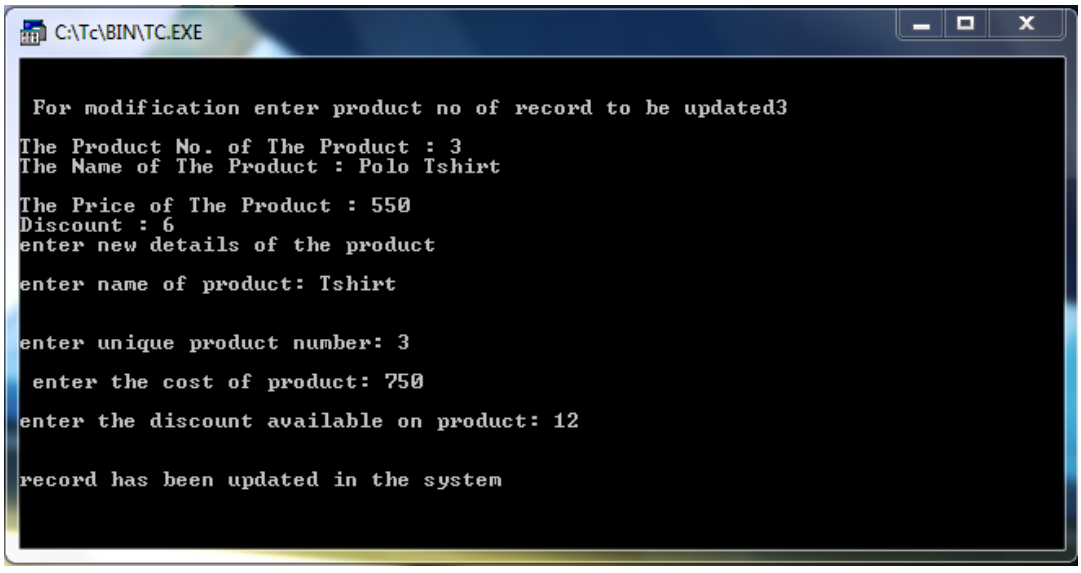
Showing all existing products



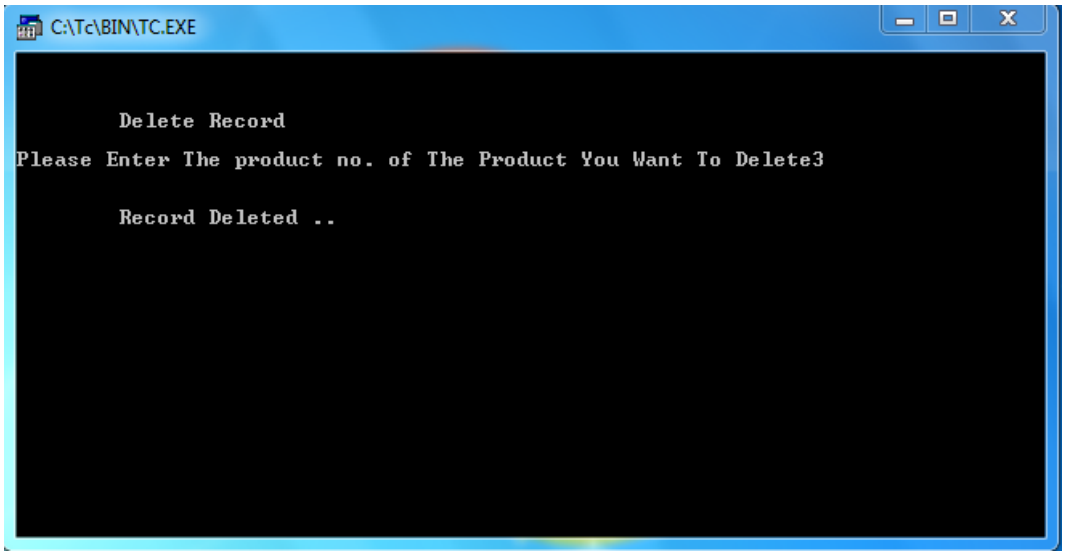
Displaying product data by product number



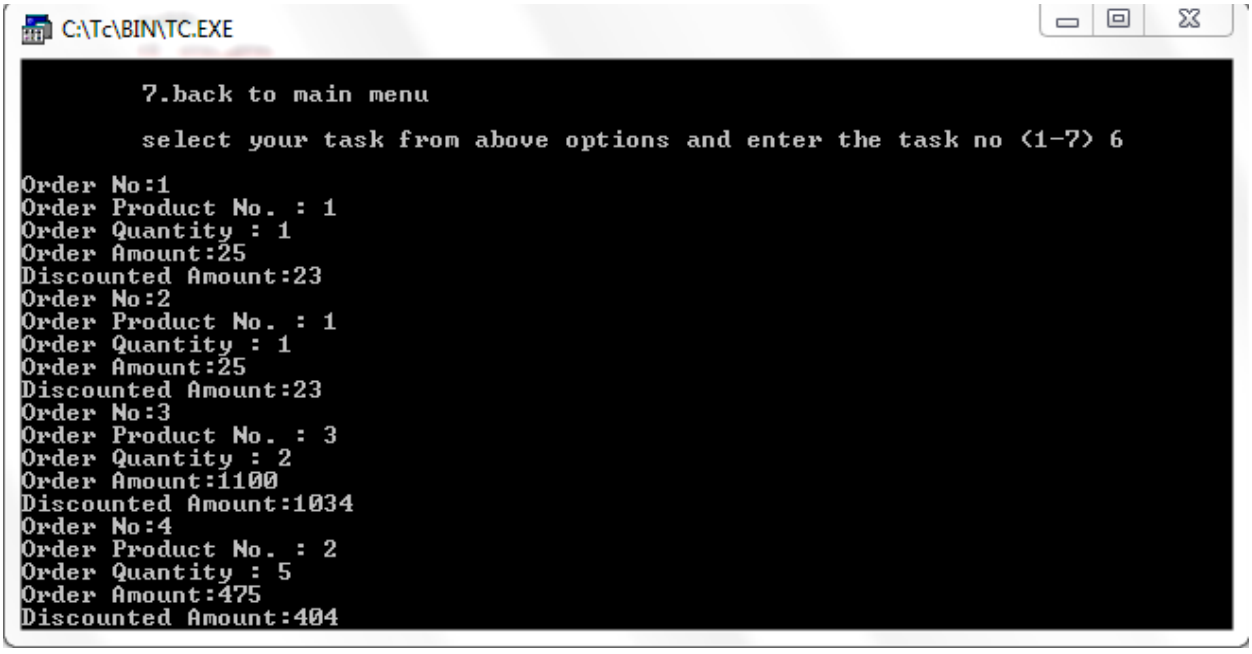
Modifying product details



Deleting product from database



Viewing order details



Ordering items