****

**POINT OF SALE**

**Group Member:-**

Muhammad Hamza (Group Leader) 21k-3815

Huzaifa Awan 21k-3835

Mohammad Zain Khan 21k-3828

Areeba Abid 21k-3904

**Project report**

**Point of sale**

Introduction:

Point of sale system, allows efficient way of checkouts at super markets. The program manages the market and assists customers to buy groceries and items of their need with minimum efforts. it is modern, faster, precise and user-friendly way of managing a supermarket and as there is no human involvement, it cancels the possibility of a human error.

Methods:

Project is made on filing system, and various other methods available in C language are used to make the code generic and user friendly. Firstly, pointers not only to create file but for multiple other purposes. Secondly, structures have been used to store the different kinds of information of the items available in the store. Thirdly, dynamic memory allocation is used to allocate the memory or the information stored in the file to a variable or a structure for calculations and other purposes. Lastly, methods like array, loops, functions, etc. have been used in multiple locations and for multiple purposes.

Functionalities:

Project manages inventory for the owner and restocks the inventory whenever items run out of quantity. Moreover, it asks user the items they want to buy and displays the available inventory so that they may choose from it. And as the customer enters the code of a particular item it automatically reduces the quantity of that particular item by the quantity they have asked for. After customers have made their grocery list, program asks from them that whether they want to add, delete or change the quantity of any item they have entered before providing them their bill. Customers can also ask for the number of items in the inventory right now. Moving on, a function of admin is also available which asks for an id and password for security reasons as admin can only be accessed by the management of the supermarket. Inside the function has options to update inventory, delete, append an item in the inventory. Admin can even search for any item available in the inventory.

User:

There can be only three users of the program: admin (management of the supermarket), customers, cashier.

Code:-

#include<Stdio.h>

#include<Conio.h>

#include<stdlib.h>

#include<string.h>

#include<windows.h>

typedef struct inv{

int no;

char it[20];

int quan ;

int pr\_per;

}inv;

void loadingBar()

{

// 0 - black background,

// A - Green Foreground

system("color 0A");

// Initialize char for printing

// loading bar

char a = 177, b = 219;

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

printf("\n\n\n\n\t\t\t\t\tLoading...\n\n");

printf("\t\t\t\t\t");

// Print initial loading bar

for (int i = 0; i < 26; i++)

printf("%c", a);

// Set the cursor again starting

// point of loading bar

printf("\r");

printf("\t\t\t\t\t");

// Print loading bar progress

for (int i = 0; i < 26; i=i+3) {

printf("%c", b);

// Sleep for 1 second

Sleep(1000);

}

}

void create(){

struct inv \*it;

FILE \*fp;

int n=15,j=0,i=0;

it = (inv\*)calloc(n,sizeof(inv));

fp = fopen("Inventory.txt","w+");

for(i=0;i<n;i++){

scanf("%d",&it[i].no);

fflush(stdin);

gets(it[i].it);

scanf("%d",&it[i].quan);

scanf("%d",&it[i].pr\_per);

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

}

fclose(fp);

}

void count(){

struct inv i1;

FILE \*fp;

int i =0;

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

fseek(fp,0,SEEK\_END);

i = ftell(fp)/sizeof(inv);

printf("No of Records = %d \n",i);

fclose(fp);

}

void count\_1(){

struct inv i1;

FILE \*fp;

int i =0;

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

fseek(fp,0,SEEK\_END);

i = ftell(fp)/sizeof(inv);

printf("No of Items in basket = %d \n",i);

fclose(fp);

}

void display(){

struct inv i1 ;

FILE \*fp ;

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

printf("Item no Item name Quantity price");

while(fread(&i1,sizeof(inv),1,fp)){

printf("\n%-5d%-20s%-5d%-5d",i1.no,i1.it,i1.quan,i1.pr\_per);

}

fclose(fp);

}

void del (){

int found =0,i\_no=0;

struct inv i1;

FILE \*fp , \*fp1;

printf("Enter the item no to delete :");

scanf("%d",&i\_no);

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

fp1 = fopen("temp.txt","w");

while(fread(&i1,sizeof(inv),1,fp)){

if(i1.no == i\_no){

found =1;

}

else {

fwrite(&i1,sizeof(inv),1,fp1);

}

}

fclose(fp);

fclose(fp1);

if(found == 1){

fp = fopen("Inventory.txt","w");

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

while(fread(&i1,sizeof(inv),1,fp1)){

fwrite(&i1,sizeof(inv),1,fp);

}

fclose(fp);

fclose(fp1);

}

else {

printf("\nItem not found");

}

}

void update(){

int found=0,i\_no=0;

struct inv i1;

FILE \*fp , \*fp1;

printf("\nEnter the item no to update ");

scanf("%d",&i\_no);

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

fp1 = fopen("temp.txt","w");

while(fread(&i1,sizeof(inv),1,fp)){

if(i1.no == i\_no){

found = 1;

i1.quan = 0;

i1.pr\_per =0;

printf("\nEnter the new quantity");

scanf("%d",&i1.quan);

printf("\nEnter new price ");

scanf("%d",&i1.pr\_per);

}

fwrite(&i1,sizeof(inv),1,fp1);

}

fclose(fp);

fclose(fp1);

if (found==1){

fp = fopen("Inventory.txt","w");

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

while(fread(&i1,sizeof(inv),1,fp1)){

fwrite(&i1,sizeof(inv),1,fp);

}

fclose(fp);

fclose(fp1);

}

else{

printf("\nItem not found ");

}

}

void append(){

struct inv \*in ,i1;

FILE \*fp,\*fp1;

int n=0,i=0,found= 1,num=0;

printf("\nEnter how many items do you want to add ");

scanf("%d",&n);

in = (inv\*)calloc(n,sizeof(inv));

fp = fopen("Inventory.txt","a");

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

//while(fread(&i1,sizeof(inv),1,fp1)){

for(i=0;i<n;i++){

in[i].quan = 0;

in[i].pr\_per = 0;

printf("\nEnter the item no ");

scanf("%d",&in[i].no);

while(found == 1){

while(fread(&i1,sizeof(inv),1,fp1)){

if(in[i].no == i1.no){

printf("\nEnter uniques number ");

scanf("%d",&in[i].no);

//i1.no =1;

found = 1;

rewind(fp1);

}

else {

found = 0;

}

}

}

if(found !=1){

fflush(stdin);

printf("\nEnter the item name ");

gets(in[i].it);

printf("\nEnter the quantity ");

scanf("%d",&in[i].quan);

printf("\nEnter the price ");

scanf("%d",&in[i].pr\_per);

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

}

}

//}

fclose(fp);

fclose(fp1);

}

void search(){

int found =0;

struct inv i1;

FILE \*fp;

int ino;

printf("\nEnter the item no to search ");

scanf("%d",&ino);

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

while(fread(&i1,sizeof(inv),1,fp)){

if(i1.no == ino){

found =1;

printf("\n%-5d%-20s%-5d%-5d",i1.no,i1.it,i1.quan,i1.pr\_per);

}

}

fclose(fp);

if(found == 0){

printf("\nItem not found.........\n");

}

}

void add\_item(){

struct inv \*i , i1,i2;

int num =0,in=0;

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

printf("\nEnter how many items do you want to buy ");

scanf("%d",&num);

int q[num] , pr=0;

int found = 0,i\_no[num];

i = (inv\*)calloc(num,sizeof(inv));

fp = fopen("UserItems.txt","w");

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

for(in=0;in<num;in++){

printf("\nEnter the item no ");

scanf("%d",&i\_no[in]);

printf("\nEnter the Quantity ");

scanf("%d",&q[in]);

}

while(fread(&i1,sizeof(inv),1,fp1)){

for(in=0;in<num;in++){

if(i1.no == i\_no[in]){

i[in].no = i1.no;

strcpy(i[in].it ,i1.it);

i[in].quan = q[in];

i[in].pr\_per = (i1.pr\_per\*q[in]);

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

}

}

//fprintf(fp,"\n%-5d%-20s%-5d%-5d",i[in].no,i[in].it,i[in].quan,i[in].pr\_per);

}

fclose(fp);

fclose(fp1);

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

fp2 = fopen("Temp.txt","w");

while(fread(&i2,sizeof(inv),1,fp1)){

for(in=0;in<num;in++){

if(i2.no == i\_no[in]){

found = 1;

i2.no = i\_no[in];

strcpy(i2.it,i2.it);

i2.quan = i2.quan - q[in];

i2.pr\_per = i2.pr\_per;

}

}

fwrite(&i2,sizeof(inv),1,fp2);

}

fclose(fp1);

fclose(fp2);

if(found == 1){

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

fp2 = fopen("Inventory.txt","w");

while(fread(&i2,sizeof(inv),1,fp1)){

fwrite(&i2,sizeof(inv),1,fp2);

}

fclose(fp1);

fclose(fp2);

}

}

void inven(){

struct inv i2;

FILE \*fp , \*fp1;

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

fp1 = fopen("Temp2.txt","w");

while(fread(&i2,sizeof(inv),1,fp)){

if(i2.quan < 30 ){

printf("\nRecord Updated of %s",i2.it);

i2.no = i2.no;

strcpy(i2.it,i2.it);

i2.quan = i2.quan+50;

i2.pr\_per = i2.pr\_per;

fwrite(&i2 , sizeof(inv),1,fp1);

}

else{

fwrite(&i2,sizeof(inv),1,fp1);

}

}

fclose(fp);

fclose(fp1);

fp = fopen("Inventory.txt","w");

fp1 = fopen("Temp2.txt","r");

while(fread(&i2,sizeof(inv),1,fp1)){

fwrite(&i2,sizeof(inv),1,fp);

}

fclose(fp);

fclose(fp1);

}

void del\_item(){

int found = 0, found\_1=0 , found\_3=0, quan = 0, i\_no = 0,price=0;

struct inv \*i ,i1 , i2 ;

FILE \*fp , \*fp1 , \*fp2 , \*fp3;

printf("\nEnter the item no to moidfy ");

scanf("%d",&i\_no);

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

fp1 = fopen("temp.txt","w");

while(fread(&i1,sizeof(inv),1,fp)){

if(i1.no == i\_no){

found = 1;

while(found\_1 != 1){

printf("\nEnter the quantity to remove ");

scanf("%d",&quan);

if(i1.quan - quan >=0){

price = i1.pr\_per/i1.quan;

found\_1 = 1;

i1.no = i1.no;

strcpy(i1.it,i1.it);

i1.quan = i1.quan - quan;

i1.pr\_per = i1.pr\_per - (quan \* price) ;

fwrite(&i1,sizeof(inv),1,fp1);

fp2 = fopen("Inventory.txt","r");

fp3 = fopen("Temp1.txt","w");

while(fread(&i2,sizeof(inv),1,fp2)){

if(i2.no == i\_no ){

found\_3 =1;

i2.no = i2.no;

strcpy(i2.it,i2.it);

i2.quan = i2.quan + quan;

i2.pr\_per = i2.pr\_per;

fwrite(&i2,sizeof(inv),1,fp3);

}

else {

fwrite(&i2,sizeof(inv),1,fp3);

}

}

fclose(fp2);

fclose(fp3);

}

else{

printf("Enter the correct quantity ");

}

}

}

else{

fwrite(&i1,sizeof(inv),1,fp1);

}

}

fclose(fp);

fclose(fp1);

//fclose(fp2);

if(found == 1){

fp = fopen("UserItems.txt","w");

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

while(fread(&i1,sizeof(inv),1,fp1)){

fwrite(&i1,sizeof(inv),1,fp);

}

if(found\_3==1){

fp2 = fopen("Temp1.txt","r");

fp3 = fopen("Inventory.txt","w");

while(fread(&i2,sizeof(inv),1,fp2)){

fwrite(&i2,sizeof(inv),1,fp3);

}

}

fclose(fp3);

fclose(fp2);

fclose(fp);

fclose(fp1);

}

else{

printf("\nItem not found........\n");

}

}

void display1(){

struct inv i1 ;

FILE \*fp ;

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

printf("Item no. Item name Quantity Price");

while(fread(&i1,sizeof(inv),1,fp)){

printf("\n%-5d%-20s%-5d%-5d",i1.no,i1.it,i1.quan,i1.pr\_per);

}

fclose(fp);

}

void display\_2(){

struct inv i1 ;

double num =0;

char name[30];

int to\_quan =0 , to\_pr=0;

float gst = 0.13,total=0 ,amount =0 ,channge =0 ;

FILE \*fp ;

printf("\nEnter your name ");

fflush(stdin);

gets(name);

printf("\nEnter phone number ");

scanf("%Lf",&num);

system("cls");

printf("\nUSER NAME :- %s",name);

printf("\nPHONE NUMBER :- %.0Lf",num);

printf("\n\nItem no. Item name Quantity Price");

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

while(fread(&i1,sizeof(inv),1,fp)){

to\_quan = to\_quan + i1.quan;

to\_pr = to\_pr + i1.pr\_per;

printf("\n%-5d%-20s%-5d%-5d",i1.no,i1.it,i1.quan,i1.pr\_per);

}

gst = gst\*to\_pr;

total = to\_pr+gst;

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

printf("Total items = %d \nGst = %.2f \nNet payment = %d \nTotal payment = %.2f ",to\_quan,gst,to\_pr,total);

printf("\nEnter the amount ");

scanf("%f",&amount);

channge = amount - total;

printf("\nAmount = %.2f \nChange = %.2f ",amount,channge);

printf("\nTHANK YOU FOR SHOPPING");

fclose(fp);

}

void sort\_in\_file (){

struct inv \*i , i1;

FILE \*fp ;

int n , in , j ;

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

fseek(fp,0,SEEK\_END);

n = ftell(fp)/sizeof(inv);

rewind(fp);

i = (inv\*)calloc(n,sizeof(inv));

for(in=0;in<n;in++){

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

}

fclose(fp);

for(in=0;in<n;in++){

for(j=in+1;j<n;j++){

if(i[in].no>i[j].no){

i1 = i[in];

i[in] = i[j];

i[j] = i1;

}

}

}

fp = fopen("Inventory.txt","w");

for(in=0;in<n;in++){

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

}

fclose(fp);

}

void reset(){

FILE \*fp ;

fp = fopen("UserItems.txt","w");

fclose(fp);

}

int main(){

//create(); //only to create file

system("color 0A");

char choice,choice\_1;

int pass,r;

char user[20],f;

printf("\t\t\t\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*WELCOME TO THE Z.A.H.H.MART\*\*\*\*\*\*\*\*\*\*\* ");

loadingBar();

system("cls");

printf("\t\t\t\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*WELCOME TO THE Z.A.H.H.MART\*\*\*\*\*\*\*\*\*\*\* ");

do{

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

printf("\n1.Admin");

printf("\n2.Display Inventory");

printf("\n3.Add items");

printf("\n4.Delete Items");

printf("\n5.Print bill");

printf("\n6.No of items in your basket");

printf("\n7.Display Basket ");

printf("\n0.Exit");

printf("\nEnter your choice ");

fflush(stdin);

scanf("%c",&choice);

switch(choice){

case '1':

system("cls");

printf("\nEnter the user name ");

fflush(stdin);

gets(user);

printf("\nEnter the password ");

scanf("%d",&pass);

r =strcmp(user,"admin");

if(pass == 1234 && r==0){

do {

printf("\n1.Append inventory ");

printf("\n2.Delete an item ");

printf("\n3.Update inventory ");

printf("\n4.Search for item ");

printf("\n5.No of item in inventory ");

printf("\n6.Check Inventory to update ");

printf("\n0.Exit");

printf("\nEnter your choice ");

fflush(stdin);

scanf("%c",&choice\_1);

switch (choice\_1){

case '1':

system("cls");

append();

break;

case '2':

system("cls");

del();

break;

case '3':

system("cls");

update();

break;

case '4':

system("cls");

search();

break;

case '5':

system("cls");

count();

break;

case '6':

inven();

break;

case '0':

system("cls");

break;

default :

printf("\n\nEnter correct option");

break;

}

}while (choice\_1!='0');

}

else {

printf("Enter the correct username and password");

}

break;

case '2':

sort\_in\_file();

system("cls");

display();

break;

case '3':

system("cls");

add\_item();

display1();

break;

case '4':

system("cls");

del\_item();

display1();

break;

case '5':

system("cls");

display\_2();

reset();

Sleep(5000);

system("cls");

break;

case '6':

system("cls");

count\_1();

break;

case '7':

system("cls");

display1();

break;

case '0':

printf("\nAre you sure you want to exit ? ");

fflush(stdin);

scanf("%c",&f);

if(f == 'y' || f == 'Y'){

printf("\nTHANK YOU FOR SHOPPING ");

}

else if (f == 'n' || f == 'N'){

choice = 99;

system("cls");

}

else{

printf("\nEnter correct option ");

fflush(stdin);

scanf("%c",&f);

choice =99 ;

system("cls");

}

break;

default :

printf("\n\nEnter the correct option");

break;

}

}while(choice !='0');

}