**PROGRAM CODE:**

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

#include<time.h>

#include <termios.h>

#include <unistd.h>

//cashier and admin menu fns

//main fn

//global declaration of file pointer

//function to display stock

//function to update existing stock by name and barcode

//fn to alert

//fn to input

/\* reads from keypress, doesn't echo \*/

int getch(void)

{

struct termios oldattr, newattr;

int ch;

tcgetattr( STDIN\_FILENO, &oldattr );

newattr = oldattr;

newattr.c\_lflag &= ~( ICANON | ECHO );

tcsetattr( STDIN\_FILENO, TCSANOW, &newattr );

ch = getchar();

tcsetattr( STDIN\_FILENO, TCSANOW, &oldattr );

return ch;

}

typedef struct

{

int dt;

int mth;

int yr;

}date;

typedef struct

{

long code; //new edit

char pdt\_name[50];

char category[50];

char sub\_category[50];

float MRP;

date mfg;

date exp;

int initial\_qty; //will not be changed while pdt is sold //will be updated when stocks are updated

int qty; //current availability //will be decremented when pdt is sold //will be updated when stocks are updated

int qty\_sold; //will be incremented when pdt is sold

int qtty; //for bill purposes

}product;

product item, arr[100];

typedef struct

{

int sno;//to mintain tot no of different products

float gst;

time\_t T;

struct tm tm;

float amt;

product \*ptr;

char cname[30];//to store the cashier name under which the transsaction has happened

float tax;

}bill;

FILE \*fp;

void alphasort(product alpha[] ,int n) // Sorting strings using bubble sort

{ product temp; //a temporary variable of type product --structure

fp=fopen("product.bin","wb");

for (int i=0; i<n-1; i++)

{

for (int j=0; j<n-1-i; j++)

{

if (strcmp(alpha[j].pdt\_name,alpha[j+1].pdt\_name) > 0)

{

temp=alpha[j];

alpha[j]=alpha[j+1];

alpha[j+1]=temp;

}

}

}

fwrite(alpha,sizeof(item),n,fp);

fclose(fp);

}

void qtysort(product arr[],int n) // Sorting strings using bubble sort

{ product temp, qnt[n]; //array of structs qty

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

qnt[i]=arr[i];

for (int i=0; i<n-1; i++)

{

for (int j=0; j<n-1-i; j++)

{

if (qnt[j].qty\_sold>qnt[j+1].qty\_sold)

{

temp=qnt[j];

qnt[j]=qnt[j+1];

qnt[j+1]=temp;

}

}

}

printf("\t\tS.No\tBARCODE\t\tNAME\t\t INITIAL QUANTITY QUANTITY SOLD\n");

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

printf("\t\t%-3d\t%-4ld\t %-25s\t%4d\t \t%6d\n",i+1,qnt[i].code,qnt[i].pdt\_name,qnt[i].initial\_qty,qnt[i].qty\_sold);

getch();

}

void readnstore(char type[]) //fn to read the file and store in an array of structures

{

int lp,cnt;

product arr[100]; //array of structures //arr is an array of type product //product is a strucure

fp=fopen("product.bin","rb");

fseek(fp,0,SEEK\_END);

lp=ftell(fp); //lp-last position

rewind(fp);

cnt=lp/sizeof(item);

fread(arr,sizeof(item),cnt,fp);

fclose(fp);

if (strcmp(type,"ALPHA")==0)

alphasort(arr,cnt);

else

qtysort(arr,cnt);

}

void product\_display() //fn to display details of a particular product

{ int choice,i=0;

long c; //new edit

char x, pdt[50];

fp=fopen("product.bin","rb+");

system("clear");

printf("\n\n\t\tDETAILS OF A PRODUCT");

do

{ printf("\n\n\tPress P to enter barcode / Press N to enter product name: ");

scanf(" %c",&x);

if(x=='P' || x=='p')

{ printf("\n\n\tEnter the product barcode: ");

scanf("%ld",&c);

printf("\n\n\tThe stock details for the product with barcode %ld are: \n\n",c);

printf("\t\tS.No\tBARCODE\t\tNAME\t\t MRP\t INITIAL QUANTITY QUANTITY AVAILABLE\n\n");

rewind(fp);

while(fread(&item,sizeof(item),1,fp))

{

if(item.code==c)

{

printf("\t\t%-3d\t%-4ld\t %-25s %-6.2f\t %4d\t \t %6d\n",++i,item.code,item.pdt\_name,item.MRP,item.initial\_qty,item.qty);

}

} //end while fread

}

else if (x=='N' || x=='n')

{ getch();

printf("\n\n\tEnter the product name: ");

gets(pdt);

printf("\n\n\tThe stock details for the product %s are: \n\n",pdt);

printf("\t\tS.No\tBARCODE\t\tNAME\t\t MRP\t INITIAL QUANTITY QUANTITY AVAILABLE\n\n");

rewind(fp);

while(fread(&item,sizeof(item),1,fp))

{

if(strcmp(item.pdt\_name,pdt)==0)

{

printf("\t\t%-3d\t%-4ld\t %-25s %-6.2f\t %4d\t \t %6d\n",++i,item.code,item.pdt\_name,item.MRP,item.initial\_qty,item.qty);

}

} //end while fread

}

printf("\n\n\tPress 1 to display details of another product / Press 0 to return ");

scanf("%d",&choice);

}while(choice==1);

fclose(fp);

printf("\n\n");

}

void display\_all\_pdt()

{ int i=0;

fp=fopen("product.bin","rb+");

system("clear");

printf("\n\n\t\t\tDETAILS OF PRODUCTS");

printf("\n\n\tThe stock details of all products are: \n\n");

printf("\t\tS.No\tBARCODE\t\tNAME\t\t MRP\t INITIAL QUANTITY QUANTITY AVAILABLE\n\n");

rewind(fp);

while(fread(&item,sizeof(item),1,fp))

{

printf("\t\t%-3d\t%-4ld\t %-25s %-6.2f\t %4d\t \t %6d\n",++i,item.code,item.pdt\_name,item.MRP,item.initial\_qty,item.qty);

} //end while fread

fclose(fp);

printf("\n\n");

}

void display\_categ()

{ int choice,i=0;

char categ[50];

fp=fopen("product.bin","rb+");

system("clear");

printf("\n\n\t\tDETAILS OF PRODUCTS-CATEGORY WISE");

do

{ getch();

printf("\n\n\tEnter the category name: ");

gets(categ);

printf("\n\n\tThe stock details for the given category %s are: \n",categ);

printf("\t\tS.No\tBARCODE\t\tNAME\t\t MRP\t INITIAL QUANTITY QUANTITY AVAILABLE\n\n");

rewind(fp);

while(fread(&item,sizeof(item),1,fp))

{

if(strcmp(item.category,categ)==0)

{

printf("\t\t%-3d\t%-4ld\t %-25s %-6.2f\t %4d\t \t %6d\n",++i,item.code,item.pdt\_name,item.MRP,item.initial\_qty,item.qty);

}

} //end while fread

printf("\n\n\tPress 1 to display details of another category? / Press 0 to return ");

scanf("%d",&choice);

}while(choice==1);

printf("\n");

fclose(fp);

}

void display\_subcateg()

{ int choice,i=0;

char sub\_categ[50];

fp=fopen("product.bin","rb+");

system("clear");

printf("\n\n\t\t\tDETAILS OF PRODUCTS- SUB-CATEGORY WISE");

do

{ getch();

printf("\n\n\tEnter the sub-category name: ");

gets(sub\_categ);

printf("\n\n\tThe stock details for the given sub-category %s are: \n",sub\_categ);

printf("\t\tS.No\tBARCODE\t\tNAME\t\t MRP\t INITIAL QUANTITY QUANTITY AVAILABLE\n\n");

rewind(fp);

while(fread(&item,sizeof(item),1,fp))

{

if(strcmp(item.sub\_category,sub\_categ)==0)

{

printf("\t\t%-3d\t%-4ld\t %-25s %-6.2f\t %4d\t \t %6d\n",++i,item.code,item.pdt\_name,item.MRP,item.initial\_qty,item.qty);

}

} //end while fread

printf("\n\n\tPress 1 to display details of another sub-category / Press 0 to return ");

scanf("%d",&choice);

}while(choice==1);

printf("\n");

fclose(fp);

}

//fn to display pdts by qty sold

void display\_qtysold()

{

readnstore("QTY");

}

//function to alert(will be triggered only when login is admin)

// check if the stock is below the level

void alert()

{

int level=11;

fp=fopen("product.bin","wb+");

while(fread(&item,sizeof(item),1,fp))

{

if(item.qty<level)

{ printf("\n\n\t\t\tWARNING!\n\t\tOnly %d nos of %s is left\n\n",item.qty,item.pdt\_name);

getch();

}

}

fclose(fp);

}

void stock\_display()

{ int choice,choice2;

do

{ system("clear");

printf("\n\n\t\tDISPLAY MENU\n\n");

printf("\t1.Display details of a particular product\n");

printf("\t2.Display details of all products\n");

printf("\t3.Display details:- category wise\n");

printf("\t4.Display details:- sub-category wise\n");

printf("\t5.Display stocks by quantity sold\n") ;

printf("\t6.Go back to previous menu\n\n");

printf("\tPlease select an option(1-6): ");

scanf("%d",&choice);

switch(choice)

{

case 1:

product\_display();

break;

case 2:

display\_all\_pdt();

break;

case 3:

display\_categ();

break;

case 4:

display\_subcateg();

break;

case 5:

display\_qtysold();

break;

case 6:

choice2=0; //will to go back to admin menu

break;

}

if(choice!=6)

{ printf("\n\tPress 1 to go back to display menu / Press 0 to return ");

scanf("%d",&choice2);

}

}while(choice2==1);

printf("\n\n");

}

//function to update existing stock by name and barcode(only admin can call fn)

void update\_stock()

{

int n,q,i,choice;

long c;

char x,name[20];

fp=fopen("product.bin","rb+");

system("clear");

do

{

printf("\n\n\t\t UPDATE STOCK MENU\n\n");

i=0;

printf("\tEnter no of items: ");

scanf("%d",&n);

while (i<n)

{ printf("\n\tPress P to enter barcode / Press N to enter product name: ");

scanf(" %c",&x);

if(x=='P' || x=='p')

{

printf("\n\tEnter barcode for product %d: ",i+1);

scanf("%ld",&c);

rewind(fp); //new edit

while(fread(&item,sizeof(item),1,fp))

{ if(c==item.code)

{ fseek(fp,-sizeof(item),SEEK\_CUR);

printf("\tEnter the quantity: ");

scanf("%d",&q);

item.qty+=q;

item.initial\_qty+=q;

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

printf("\n\tProduct with barcode %ld stock updated\n",c);

break;

}

}//end while read

i++;

}//end if x==p

else if(x=='N' || x=='n')

{ getch();

printf("\n\tEnter name of the product %d: ",i+1);

gets(name);

rewind(fp); //new edit

while(fread(&item,sizeof(item),1,fp))

{ if(strcmp(name,item.pdt\_name)==0)

{ fseek(fp,-sizeof(item),SEEK\_CUR);

printf("\tEnter the quantity: ");

scanf("%d",&q);

item.qty+=q;

item.initial\_qty+=q;

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

printf("\n\tProduct %s stock updated\n",name);

break;

}

}//end while read

i++; //INVALID CHOICE TAKEN CARE //will be incremented only if the choice is correct

}//end if x==n

else

printf("\n\tYou have entered an invalid choice\n\t\tPlease enter again\n");

} //end while i<n

printf("\n\tDo you want to update stock for more items?\n\n");

printf("\tPress 1 to update stock for more items / Press 0 to return ");

scanf("%d",&choice);

}while (choice==1);

fclose(fp);

}

//function to store pdt details or to add new pdt to supermarket(only admin can call fn)

void add\_pdt() //changed fn name from input to add\_pdt!!

{

int n,choice;

srand(time(0)); // num = (rand() % (hi\_num - low\_num)) + low\_num;

fp=fopen("product.bin","ab");

do

{ system("clear");

printf("\n\n\t\tADD PRODUCT MENU\n\n");

printf("\tEnter no of items: ");

scanf("%d",&n);

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

{ printf("\n\tEnter details for product %d-\n",i+1);

item.code=(rand() % (60000 - 10000)) + 10000; //generates a five digit random no in the range 10k -90k //new edit

printf("\tEnter product category: ");

getchar();

gets(item.category);

printf("\tEnter product sub-category: ");

gets(item.sub\_category);

printf("\tEnter product name: ");

gets(item.pdt\_name);

printf("\tEnter MRP: ");

scanf("%f",&item.MRP);

printf("\tEnter manufacturing date: ");

scanf("%d-%d-%d",&item.mfg.dt,&item.mfg.mth,&item.mfg.yr);

printf("\tEnter expiry date: ");

scanf("%d-%d-%d",&item.exp.dt,&item.exp.mth,&item.exp.yr);

printf("\tEnter qty: ");

scanf("%d",&item.initial\_qty);

item.qty=item.initial\_qty;

item.qty\_sold=0;

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

}

printf("\n\tDo you want to add more items?\n\n");

printf("\tPress 1 to add more items / Press 0 to return ");

scanf("%d",&choice);

}while(choice==1);

fclose(fp);

readnstore("ALPHA");

}

//to remove a product from the file

void delete()

{

char ch,pname[50];

FILE \*rfp,\*wfp;

int pcode,opt;

system("clear");

printf("\n\n\t\t\tDELETE MENU\n\n");

getchar();

printf("\tDo you want to delete an existing product in the list? (Y/N) : ");

scanf(" %c",&ch);

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

{ do

{

do

{

printf("\n\tPress 1 to delete using product code / Press 2 to delete using product name ");

scanf("%d",&opt);

switch(opt)

{

case 1 :

printf("\n\tEnter the code of the product to be deleted : ");

scanf("%d",&pcode);

wfp=fopen("new.bin","wb");//create and append the data

rfp=fopen("product.bin","rb");//open in read and write mode with file ptr at the beginning

if(!rfp||!wfp)

puts("\n\tError!!file can't be opened!\n");

else

{

int flag=0;

while(fread(&item,sizeof(item),1,rfp))

{

if(item.code==pcode)

flag=1;

else

fwrite(&item,sizeof(item),1,wfp);

}//while fread close

if(flag==1)

{ printf("\n\tProduct with code %d deleted",pcode);

remove("product.bin");

rename("new.bin","product.bin");

}//new file with product deleted is renamed with old file only if there is a difference in data between the two files

else

puts("\n\tOOPS!THERE IS NO PRODUCT WITH THE ENTERED CODE :(\n");

}//else close

fclose(rfp);

fclose(wfp);

break;

case 2 :

getchar();

printf("\n\tEnter the name of the product to be deleted : ");

gets(pname);

wfp=fopen("new.bin","wb");//create and append the data

rfp=fopen("product.bin","rb");//open in read and write mode with file ptr at the beginning

if(!rfp||!wfp)

puts("\n\tError!!file can't be opened!\n");

else

{

int flag=0;

while(fread(&item,sizeof(item),1,rfp))

{

if(strcmp(pname,item.pdt\_name)==0)

flag=1;

else

fwrite(&item,sizeof(item),1,wfp);

}//while fread close

if(flag==1)

{ printf("\n\tProduct %s deleted",pname);

remove("product.bin");

rename("new.bin","product.bin");

}//new file with product deleted is renamed with old file only if there is a difference in data between the two files

else

puts("\n\tOOPS!!THERE IS NO PRODUCT WITH THE GIVEN NAME!\n");

}//else close

fclose(rfp);

fclose(wfp);

break;

default :

puts("\n\tInvalid choice!! enter your choice again ");

}//switch close

}while(opt!=1 && opt!=2);

getchar();

printf("\n\n\tDo you want to delete another product from the existing list (Y/N) : ");

scanf(" %c",&ch);

}while(ch=='y'||ch=='Y');

}//if ch=y or Y close

}//delete close

void adminmenu(char username[9])

{ int choice,choice2;

//alert();

do

{ system("clear");

printf("\n\n\t\tWelcome %s\n\n",username);

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

printf("\t1.View stock details\n");

printf("\t2.Update the stock for a product\n");

printf("\t3.Add a product to database\n");

printf("\t4.Remove a product from database\n");

printf("\t5.Log out\n\n"); // Go back to login page

printf("\tPlease select an option(1-6): ");

scanf("%d",&choice);

switch(choice)

{

case 1:

stock\_display();

break;

case 2:

update\_stock();

break;

case 3:

add\_pdt();

break;

case 4:

delete();

break;

case 5:

choice2=0; //will to go back to login page

break;

}

if(choice!=5)

{ printf("\n\n\tPress 1 to go back to admin menu / Press 0 to log out ");

scanf("%d",&choice2);

}

}while(choice2==1);

}

//insert fn to view sales/bill details here!

void print\_bill(int flag,bill b)

{

system("clear");

printf("\n\n\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*BILL\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n\n");

if(flag==1)

{

FILE \*bfp;

bfp=fopen("bill.bin","ab+");

if(feof(bfp))

puts("\n\n\t\tSorry! Error in generating bill");

else

{

fwrite(&b,sizeof(b),1,bfp);

fclose(bfp);

}//writing bill history into a file named bill

//print bill data after writing nto file for customer's copy

printf("\n\t\t\t XYZ SUPERMARKET\n\t\tThe one-stop place for all your needs!\n\n");

printf("\n\n\t\tDate : %02d/%02d/%04d\t",b.tm.tm\_mday, b.tm.tm\_mon+1, b.tm.tm\_year+1900);

printf("Time : %02d:%02d:%02d\n",b.tm.tm\_hour, b.tm.tm\_min, b.tm.tm\_sec);

printf("\n\t\tSNO ITEM PRICE QTY AMOUNT \n" );

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

for(int i=0;i<b.sno;i++)

{

printf("\n\n\t\t %-2d %-20s",i+1,(b.ptr+i)->pdt\_name);

float t=(b.ptr+i)->qtty\*(b.ptr+i)->MRP;

printf(" %-6.2f %-4d %-5.2f",(b.ptr+i)->MRP,(b.ptr+i)->qtty,t);

printf("\n");

}

printf("\n\n\t\t\t\t\t\tTOTAL AMOUNT : %.2f\n",b.amt);

}//outer if

else

puts("\n\n\t\tNo valid items added in the bill ! Thus failed to generate:(\n\n");

}

void bill\_input(char username[])

{

int i=1;

product \*temp=NULL;

int flag=0;

bill b;

char ch;

strcpy(b.cname,username);//cashier name

b.T= time(NULL);

b.tm = \*localtime(&b.T);//date and time

b.amt=0;

system("clear");

printf("\n\n\t\t\tBILL MENU\n\n");

do

{

getchar();

printf("\n\t\tPlease enter product name :");

char str[50];

gets(str);//prod name

str[strlen(str)]='\0';

FILE \*fp;

fp=fopen("product.bin","rb+");

product obj;

do

{

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

}while((strcmp(obj.pdt\_name,str)!=0)&&(feof(fp)==0));

if(strcmp(str,obj.pdt\_name)!=0)

{

printf("\n\n\toops,entered product is not available in the stock.\nproduct name is case sensitive, check the spelling!!\n");

}

else

{

b.ptr=temp;

b.sno=i;//serial no

//dynamic allocation of ptr for a single product in the bill - pointer object of type bill

if(i==1) (b.ptr)=(product \*)calloc(i,sizeof(product));

else (b.ptr)=(product \*)realloc(b.ptr,i\*sizeof(product));

temp=b.ptr;

b.ptr+=(i-1);

strcpy((b.ptr)->pdt\_name,str);

\*(b.ptr)=obj;//ptr points to the object with the matching product name

printf("\n\n\t\tquantity : ");

scanf("%d",&(b.ptr)->qtty);//prod quantity

(b.ptr)->qty-=1;//availability

(b.ptr)->qty\_sold+=1;//no. purchased

fseek(fp,-sizeof(obj),SEEK\_CUR);//moving file pointer back by obj bytes

fwrite((b.ptr),sizeof(obj),1,fp);//modifying the product quantity as per the purchase

b.amt+=((b.ptr)->MRP\*(b.ptr)->qtty);

flag=1;

i++;

}//else closes

getchar();

printf("\n\n\t\tDo you want to add another product to the bill?(Y/N) : ");

scanf(" %c",&ch);

fclose(fp);

}while(ch=='y'||ch=='Y');

char choice;

b.tax=0.18;

b.amt=b.amt+(b.tax\*(b.amt));

getchar();

printf("\n\t\tDo you want to proceed in generating the bill (Y/N) : ");

scanf(" %c",&choice);

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

{

b.ptr=temp;

print\_bill(flag,b);

}

else

puts("\n\t\tBill failed to generate! :(");

b.ptr=temp;

free(b.ptr);

}//input() close

void cashiermenu(char username[9])

{ int choice,choice2;

do

{ system("clear");

printf("\n\n\t\tWelcome %s\n\n",username);

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

printf("\t1.Generate bill for a customer\n"); //cashier can only generate bill

printf("\t2.Log out\n\n"); // Go back to login page

printf("\tPlease select an option(1-2): ");

scanf("%d",&choice);

switch(choice)

{

case 1:

bill\_input(username); //new edit

break;

case 2:

choice2=0; //will to go back to login page

break;

}

if(choice!=2)

{ printf("\n\n\tPress 1 to go back to cashier menu / Press 0 to log out ");

scanf("%d",&choice2);

}

}while(choice2==1);

}

void main()

{

char adminuser[]="admin001", adminpass[]="abc123";

char cashieruser1[]="cashier1",cashierpass1[]="xyz123"; //all usernames are of len 8 and passwords are of len 6

char username[strlen(adminuser)],password[strlen(adminpass)];

int choice,choice2,i;

char ch;

do

{

system("clear");

printf("\n\n\n\t\t\tWelcome to XYZ SUPERMARKET\n\n"); //home page

printf("\t\t\t\tLOGIN AS:\n\n\t\t1.ADMINISTRATOR\t\t2.CASHIER\n\n\t\t\t\t3.EXIT\n\n\t\tPlease select an option(1-3): ");

scanf("%d",&choice);

if(choice==1)

{ do

{ system("clear");

printf("\n\n\n\t\t\tADMINISTRATOR LOGIN\n\n");

printf("\tEnter your id: ");

scanf("%s",username);

getch();

printf("\n\tEnter your password: ");

for(i=0;i<strlen(adminpass);i++)

{

ch = getch();

password[i] = ch;

ch = '\*' ;

printf("%c",ch);

}

password[i] = '\0';

getch();

if ( strcmp(password,adminpass)!=0 || strcmp(username,adminuser)!=0)

{ printf("\n\n\tInvalid username or password!\n\n\tPress enter to try again!\n\n");

getch();

}

}while(strcmp(password,adminpass)!=0 || strcmp(username,adminuser)!=0);

system("clear");

adminmenu(username);

}

else if(choice==2)

{ do

{ system("clear");

printf("\n\n\n\t\t\tCASHIER LOGIN\n\n");

printf("\tEnter your id: ");

scanf("%s",username);

getch();

printf("\n\tEnter your password: ");

for(i=0;i<strlen(cashierpass1);i++)

{

ch = getch();

password[i] = ch;

ch = '\*' ;

printf("%c",ch);

}

password[i] = '\0';

getch();

if (strcmp(password,cashierpass1)!=0 || strcmp(username,cashieruser1)!=0)

{ printf("\n\n\tInvalid username or password!\n\n\tPress enter to try again!\n\n");

getch();

}

}while (strcmp(password,cashierpass1)!=0 || strcmp(username,cashieruser1)!=0);

system("clear");

cashiermenu(username);

}

else

{ printf("\n\n");

exit(0); //exits the system //why not return 0?

}

printf("\n\n\tPress 1 to go back to the login page / Press 0 to exit "); //goes to login menu

scanf("%d",&choice2);

}while(choice2==1);

printf("\n\n");

}