

ONLINE SHOPPING

_BY:

Sharjeel Ahmed (K20-0288) (leader)

Ahzam Imam (K20-1612)

Header Files:

```
#include "colorprintf.h"

#include <time.h>
#include <stdio.h>
#include <conio.h>
#include <string.h>
#include <stdlib.h>
```

Structures/Enum:

```
enum bool{false,true};
typedef struct items
{
    char productname[20];
    int productid;
    float price;
} items;
typedef struct user //for users if new (register ) otherwise provide username and
password
{
    char name[30];
    int code;
    char phone_num[11];
} user;
typedef struct order
{
    int order_no;
    int ship_day, p_id;
    float total_price;
    char c_address[40], mobile_no[12], product[25];
    time_t time;
```

```
} order;
```

Functions ProtoType:

```
void delay(int no_of_sec); //waiting unction
void admin();              //to access as admin (admin)
void inventory();          //to access inventory (admin)
int product_id();          // for assigning unique id to every product
void add_inventory();      //to add products (admin)
void del_inventory();      //to del the inventory(admin)
void edit_inventory();     // to edit specific item (admin)
void orders();             // for orders (admin)
void search_orders();      // for searching orders (admin)
void view_all_order();     // for view all orders (admin)
void billing();            // to see all the bills (admin)
void all_inventory();      //to view all inventory (admin)
void registered_user();    // to view all the registered users (admin)
void search_inventory();   // to search a product by id or name (admin)
void customer();           // to access (customer)
void new_user();           //if new user register (customer)
void old_user();           //if new user provide password (customer)
void delivery();           // to access delivery status (delivery)
void outlet();             // to show all the products (admin) (customer)
void show_bill(int code);  // for generating the slip of purchased items (customer)
int orders_no();           // for generating order no
void chatbot();
void delayprint(int no_of_sec); //delay in printing alphabets
void H();
void I();
```

Main:

```
int main()
{
    // system("color 4F"); //for color back ground
    system("cls");
    H(); //printing h
    system("cls");
```

```

I();
system("cls");
//system("color 0");
colorprintf(0, "Welcome to ZIDELLO's\n");
int num, pin_counter = 3; // to choose b/w opt
a:                                     //if in put is wrong take input again
b:                                     // if any function is exit it will run 1 more
    colorprintf(3, "Press 1 to access ADMIN portal\nPress 2 to access CUSTOMER
portal\nPress 3 for DELIVERY information\nPress 4 for Customer care\nPress 0 f
or exit\n");
    scanf("%d", &num);
    switch (num)
    {
    case 1: //admin
    {
        system("cls");
start:
        if (pin_counter >= 0) /*counter for the admin's pin*/
        {
            char str[13], c, password[] = "admin123";
            int i = 0, pass, *passptr2;
            passptr2 = &pass;

            if (pass == 8)
                goto skip;
            colorprintf(3, "enter the password\n");
            while (i <= 17) /* for printing * on every word */
            {
                c = getch();
                //str[i] = c;

                if (c == '\b')
                {
                    if (c == 13)
                        break;
                    colorprintf(3, "\b");
                    --i;
                }
                else if (c != '\b')
                {
                    if (c == 13)
                        break;
                    colorprintf(3, "*");
                    str[i] = c;
                    i++;
                }
            }
        }
    }

```

```

    }
}
str[i] = '\0';
skip:
if ((strcmp(str, password) == 0) || pass == 8)
{
    int **passptr; /*if admin input correct password then he allowed t
o access without pin (only for consoul)*/
    passptr = &passptr2;
    **passptr = 8;
    printf("\n");
    delay(4);

    system("cls");
    pin_counter = 0;
    admin();
}

else /*wron pin input*/
{
    printf("\n");
    delay(6);
    Beep(523, 500);
    colorprintf(3, "\nwrong pin\n");
    system("cls");
    colorprintf(3, "%d attemp remaining\n", pin_counter);
    pin_counter--;
    goto start;
}
}
else /*3 time wrong pin (blocked)*/
{
    delay(3);
    Beep(523, 1000);
    colorprintf(3, "your admin id is blocked\ncall 21684984651 for admin
support");
    return 0;
}
break;
}
case 2: /*to access customer*/
    customer();
    delay(5);
    break;
case 3:

```

```

        delivery();
        delay(5);
        break;
    case 4:
        chatbot();
        delay(5);
        break;
    case 0: /*for quitting*/
        system("cls");
        delay(4);
        colorprintf(3, "\nthank you for visiting/shopping from zidello's\n");
        delay(3);
        return 0;
    default:
        Beep(523, 500);
        colorprintf(3, "wrong input \n");
        delay(2);
        system("cls");
        goto a;
}
system("cls");
/* printf("0 for exit\n1 for start ");
int choose;
scanf("%d",&choose);
if(choose==1)*/
goto b;
/*else
{printf("thank you for visiting/shopping from zidello's\n");
return 0;}*/
}

```

➔ Admin has to log in once every time the program open (if he has to log in for every attribute then it will become frustration for him) by the help of double pointer we counter this problem.

ADMIN

1. Inventory

I. Search item from inventory

- a) By product name
- b) By product id

- II. Edit item in the inventory
- III. Add new item in the inventory
- IV. Delete item from the inventory
- V. View all inventory

2. Orders

- I. Search order of specific product
 - a) Search by product name
 - b) Search by product id
- II. View all orders

3. Billing (All generated Bills)

4. All registered user's data

```
void admin()
{
    int num;
a:
    colorprintf(5, "1  inventory\n2  Orders\n3  Billing\n4  Registered users\n0
for exit\n");
    scanf("%d", &num);
    switch (num)
    {
    case 1:
        inventory();
        delay(2);
        break;
```

```

    case 2:
        orders();
        delay(2);
        break;
    case 3:
        billing();
        delay(2);
        break;
    case 4:
        registered_user();
        delay(2);
        break;
    case 0:
        colorprintf(0, "**EXITING**\n");
        delay(2);
        break;
    default:
        Beep(523, 500);
        printf("wrong input \n");
        delay(2);
        system("cls");
        goto a;
}
}

```

INVENTORY:

```

void inventory()
{
a:
    system("cls");
    int num;
    colorprintf(0, "1 search item\n2 edit item\n3 add new item\n4 delete item\n5 view all inventory\n0 for exit\n");
    scanf("%d", &num);
    fflush(stdin);
    switch (num)
    {
    case 1:
        search_inventory();
        delay(3);
        break;
    case 2:
        edit_inventory();
    }
}

```

```

        delay(3);
        break;
    case 3:
        add_inventory();
        delay(3);
        break;
    case 4:
        del_inventory();
        delay(3);
        break;
    case 5:
        all_inventory();
        delay(3);
        break;
    case 0:
        colorprintf(0, "exiting\n");
        delay(3);
        break;
    default:
        Beep(523, 500);
        colorprintf(0, "wrong input\n");
        delay(3);
        goto a;
}
}

```

```

void add_inventory()
{
    FILE *shop = fopen("shop.txt", "a");
    items A;
    system("cls");
    colorprintf(3, "how many items do you want to add\n");
    int add;
    int adder = 0, counter = 0;
    scanf("%d", &add);
    fflush(stdin);
    for (int i = 0; i < add; i++)
    {

        A.productid = product_id() + (adder++);
        fflush(stdin);
    input_name:
        colorprintf(3, "enter the product name : ");
        gets(A.productname);
    }
}

```



```

        fflush(stdin);
        for (int p = 0; p < strlen(A.productname); p++)
        {
            if ((A.productname[i] >= 'a' && A.productname[i] <= 'z') || (A.productname[i] >= 'A' && A.productname[i] <= 'Z'))
                counter++;
        }
        if (counter < 3)
        {
            colorprintf(0, "\nPlease enter the valid product name\n");
            delay(3);
            system("cls");
            goto input_name;
        }

        colorprintf(3, "enter the product price : ");
        scanf("%f", &A.price);
        fflush(stdin);
        fwrite(&A, sizeof(items), 1, shop);
        colorprintf(3, "\n");
    }
    // fwrite((A + i), sizeof(*(A + i)), add, shop);

    colorprintf(0, "\n****ITEMS SUCCESFULLY ADDED***\n**PRESS ANY KEY TO CONTINUE**\n");
    getch();
    delay(2);
    fclose(shop);
}

```

```

void all_inventory()
{
    system("cls");
    items A;
    FILE *shop;
    int i = 1;
    shop = fopen("shop.txt", "r");
    if (shop == NULL)
    {
        printf("error\n");
        exit(1);
    }
}

```

```

system("cls");
printf("=====
=====\\n");
colorprintf(0, "S.NO\\tP.ID\\t\\t      P.NAME\\t\\t\\t      RS\\n");
printf("=====
=====\\n\\n");

while (fread(&A, sizeof(A), 1, shop) > 0)
{
    colorprintf(3, "%-3d\\t%-8d\\t\\t%-20s\\t%-
8.3f\\n", i++, A.productid, A.productname, A.price);
}
printf("\\n=====
=====\\n\\n");
colorprintf(0, "\\n\\n****PRESS ANY KEY TO CONTINUE****\\n");
getch();
delay(2);
fclose(shop);
}

```

```

void del_inventory()
{
    system("cls");
    FILE *shop, *rep;
    items A;
    int del_line, temp = 1, i = 1;
    shop = fopen("shop.txt", "r");
    if (shop == NULL)
    {
        shop = fopen("shop.txt", "w");
        printf(" error in opening \\n new file is just created \\n");
        exit(1);
    }
    printf("=====
=====\\n");
    colorprintf(0, "S.NO\\tP.ID\\t\\t      P.NAME\\t\\t\\t      RS\\n");
    printf("=====
=====\\n\\n");

    while (fread(&A, sizeof(A), 1, shop) > 0)
    {

```

```

        colorprintf(3, "%-3d\t%-8d\t\t%-20s\t%-
8.3f\n", i++, A.productid, A.productname, A.price);
    }
    printf("\n=====
=====\\n\\n");

    rewind(shop); //rewind ptr
    colorprintf(0, " \nEnter line number of the line to be deleted : ");
    scanf("%d", &del_line);
    rep = fopen("replica.txt", "w"); //open new file in write mode

    while (fread(&A, sizeof(items), 1, shop) > 0)
    {

        //except the line to be deleted
        if (temp != del_line)
        {
            fwrite(&A, sizeof(items), 1, rep);
            //copy all lines in file replica.c
        }
        temp++;
    }
    fclose(shop);
    fclose(rep);
    remove("replicas.txt"); //remove (file) replicas
as
    if (rename("shop.txt", "replicas.txt") == 0) //swaping file names
shop->replicas
        colorprintf(0, "\\n\\n***deleted succesfully***\\n"); //replica->shop
    else
        printf("error\\n");
    if (rename("replica.txt", "shop.txt") == 0)
        printf("");
    else
        printf("error\\n");
    if (rename("replicas.txt", "replica.txt") == 0)
        printf("");
    else
        printf("error\\n");
    remove("replicas.txt"); //remove (file) replicas
    fclose(shop);
    fclose(rep);
    colorprintf(5, "****PRESS ANY KEY TO CONTINUE***\\n");
    getch();
    delay(2);

```

```
}
```

```
void search_inventory()
{
    FILE *shop;
    shop = fopen("shop.txt", "r");
    items A;
a:
    system("cls");
    colorprintf(0, "search by\n1 product name\n2 product id\n0 for exit\n");
    int num;
    scanf("%d", &num);
    switch (num)
    {
        case 1:
        {
            system("cls");
            char pname[25];
            colorprintf(3, "enter the name of product : ");
            fgets(pname, sizeof(pname), stdin);
            gets(pname);

            while (fread(&A, sizeof(items), 1, shop) > 0)
            {
                if (strcmp(pname, A.productname) == 0)
                {
                    colorprintf(3, "\n\n%d %s %f", A.productid, A.productname, A.pr
ice);
                    break;
                }

                //else printf("error!\n");
            }
            break;
        }
        case 2:
        {
            int id;
            colorprintf(0, "enter the id of product : ");
            scanf("%d", &id);
            while (fread(&A, sizeof(A), 1, shop) > 0)
            {
                if (id == A.productid)
                {
```

```

        colorprintf(3, "\n\n%d  %s  %f", A.productid, A.productname, A.pr
ice);
        break;
    }

    // else printf("error!\n");
}
break;
}
case 0:
    colorprintf(0, "redirecting\n");
    break;
default:
    Beep(523, 500);
    colorprintf(0, "wrong input\n");
    goto a;
}
colorprintf(0, "\n\n****PRESS ANY KEY TO CONTINUE****\n");

getch();
delay(2);
fclose(shop);
}

```

Orders:

```

void orders()
{
    system("cls");
    int num;
a:
    colorprintf(0, "1  search orders\n2  view all orders\n0  For Exit\n");
    scanf("%d", &num);
    switch (num)
    {
        case 1:
            search_orders();
            delay(3);
            break;
        case 2:
            view_all_order();
            delay(3);
            break;
        case 0:

```

```
printf(0, "***EXITING***\n");
delay(3);
break;
default:
    Beep(523, 500);
    system("cls");
    printf("wrong input\n");
    delay(3);
    goto a;
}
```

```
void view_all_order()
{
    system("cls");
    FILE *outlet;
    order A;
    outlet = fopen("outlet.txt", "r");
    if (outlet == NULL)
    {
        printf("could not load \nerror");
        exit(1);
    }
    printf("=====\n");
    colorprintf(0, "PRODUCT\t\t\t PRODUCT_ID\t\t\t\t\t DATE\n");
    printf("=====\n\n");
    while (fread(&A, sizeof(order), 1, outlet) > 0)
    {
        colorprintf(3, "%-20s\t%-9d\t\t\t %s", A.product, A.p_id, ctime(&A.time));
    }
    printf("\n=====\\n\n");
    fclose(outlet);
    colorprintf(0, "PRESS ANY KEY TO CONTINUE\n");
    getch();
    delay(2);
}
```

```

void search_orders()
{
    FILE *outlet;
    outlet = fopen("outlet.txt", "r");
    order A;
    if (outlet == NULL)
    {
        printf("error\n");
        exit(1);
    }
a:
    system("cls");
    colorprintf(0, "search by\n1  product name\n2  product id\n0  for exit\n");
    int num;
    scanf("%d", &num);
    switch (num)
    {
    case 1:
    {
        system("cls");
        char pname[25];
        fflush(stdin);
        colorprintf(3, "enter the name of product : ");
        //fgets(pname, sizeof(pname), stdin);
        gets(pname);
        printf("=====
=====\\n");
        colorprintf(0, "PRODUCT\\t\\t      PRODUCT_ID\\t\\t \\t      DATE\\n");
        printf("=====
=====\\n\\n");
        while (fread(&A, sizeof(order), 1, outlet) > 0)
        {
            if (pname == A.product)
                colorprintf(3, "%-20s\\t%-
9d\\t\\t  %s", A.product, A.p_id, ctime(&A.time));
        }
        printf("\\n=====
=====\\n\\n");

        break;
    }
    case 2:
    {
        system("cls");
        int id;

```

```

        fflush(stdin);
        colorprintf(0, "enter the id of product : ");
        scanf("%d", &id);
        printf("=====
=====\\n");
        colorprintf(0, "PRODUCT\\t\\t      PRODUCT_ID\\t\\t \\t      DATE\\n");
        printf("=====
=====\\n\\n");
        while (fread(&A, sizeof(order), 1, outlet) > 0)
        {
            if (id == A.p_id)
                colorprintf(3, "%-20s\\t%-
9d\\t\\t      %s", A.product, A.p_id, ctime(&A.time));
        }
        printf("\\n=====
=====\\n\\n");

        break;
    }
    case 0:
        colorprintf(0, "redirecting\\n");
        break;
    default:
        Beep(523, 500);
        colorprintf(0, "wrong input\\n");
        delay(2);
        system("cls");
        goto a;
    }
    colorprintf(0, "****PRESS ANY KEY TO CONTINUE****\\n");

    getch();
    delay(2);
    fclose(outlet);
}

```

All bills

```

void billing()
{
    system("cls");
    FILE *outlet;

```



```

order show;
outlet = fopen("outlet.txt", "r");
if (outlet == NULL)
{
    colorprintf(0, "could not load \nerror");
    exit(1);
}
printf("=====
=====
====\n");
    colorprintf(0, "  PRICE\t\tPRODUCT\t\t  PRODUCT_ID\t  ORDER_NO    SHIP_DAY\t
\tMOBILE_NO\t      ADDRESS\t\t\t  DATE\n");
    printf("=====
=====
====\n\n");

    while (fread(&show, sizeof(order), 1, outlet) > 0)
    {

        {
            colorprintf(3, "%8.3f\t%-18s %-8d\t %-8d      %-2d\t\t%-12s %-
25s\t  %s", show.total_price, show.product, show.p_id, show.order_no, show.ship
_day, show.mobile_no, show.c_address, ctime(&show.time));
        }
    }
    printf("\n=====
=====
====\n\n");
    fclose(outlet);
    colorprintf(0, "****PRESS ANY KEY TO CONTINUE****\n");
    getch();
    delay(2);
}

```

All registered users

```

void registered_user() //to read file data (structure filing)
{
    system("cls");
    colorprintf(0, "****ALL REGISTERED USERS****\n\n");
    FILE *cstmr;
    cstmr = fopen("user.txt", "r");
}

```



```

a:
    colorprintf(0, "1  For SIGN UP\n2  For LOG IN\n0  For EXIT\n");
    scanf("%d", &num);
    switch (num)
    {
        case 1:
            system("cls");
            new_user();
            delay(3);
            break;
        case 2:
            system("cls");
            old_user();
            delay(3);
            break;
        case 0:
            system("cls");
            printf("exiting ");
            delay(3);
            break;
        default:
            Beep(523, 500);
            system("cls");
            printf("wrong input ");
            delay(2);
            system("cls");
            goto a;
    }
}

```

```

void new_user()
{
    system("cls");
    colorprintf(0, "****GREATINGS****\n");
    FILE *cstmr;
    user u, isnew;
    int check;
    fflush(stdin);
    invalidEmail:
    check=false;
    colorprintf(3, "enter your email address : ");
    gets(u.name);
}

```

```

for(int i=0;i<strlen(u.name);i++)
{
    if(u.name[i]=='@')
        check=true;
}
if(check!=true)
{
    Beep(523, 500);
    colorprintf(0,"Please enter valid email\n");
    goto invalidEmail;
}
check=false;
cstmr = fopen("user.txt", "r");
while(fread(&isnew,sizeof(isnew),1,cstmr)>0)
{
    if(strcmp(u.name,isnew.name)==0)
        check=true;
}
if(check==true)
{
    Beep(523, 500);
    system("cls");
    colorprintf(0,"This email address is already registered, please enter email again\n");
    goto invalidEmail;
}
fclose(cstmr);
//fputs(u->name, cstmr);
fflush(stdin);
input_num:
colorprintf(3, "\nenter your mobile number : ");
gets(u.phone_num);
if (strlen(u.phone_num) != 11)
{
    Beep(523, 500);
    colorprintf(0, "\nPLEASE ENTER THE VALID MOBILE NUMBER\n");
    delay(2);
    system("cls");
    goto input_num;
}
for (int i = 0; i < strlen(u.phone_num); i++)
{
    if ((u.phone_num[i] >= 'a' && u.phone_num[i] <= 'z') || (u.phone_num[i] >= 'A' && u.phone_num[i] <= 'Z'))

```

```

    {
        colorprintf(0, "\nPLEASE ENTER THE CORRECT MOBILE NUMBER\n");
        delay(2);
        system("cls");
        goto input_num;
    }
}
fflush(stdin);
colorprintf(3, "\nenter 6 digit password (first should be non-zero): ");
scanf("%d", &u.code);
fflush(stdin);

cstmr = fopen("user.txt", "a");
fwrite(&u, sizeof(user), 1, cstmr);
//fprintf(cstmr, " %d \n", u->code);
fflush(stdin);
colorprintf(0, "\nredirectiong you to the login\n");
delay(2);
fclose(cstmr);
old_user();
}

```

Key Features:

If you are trying to input only name or any other thing in email program will show an error and take input again and if email is already registered then program show a message "This Email is already registered" and takes input again.

If you enter less or more than 11 digits in number then program takes number input again beside that you are not allowed to input any other character.

```

void old_user()
{
    system("cls");
    colorprintf(0, "****PLEASE LOG IN****\n");
    FILE *cstmr;
    cstmr = fopen("user.txt", "r");
    if (cstmr == NULL)
    {
        colorprintf(0, "error\n");
    }
}

```

```

        exit(1);
    }
    user u;
    char email[25];
    int password, counter = 3;
again:
    fflush(stdin);
    colorprintf(3, "\nenter your email address : ");
    gets(email);
    fflush(stdin);
    colorprintf(3, "\nenter your six digit password : ");
    scanf("%d", &password);
    fflush(stdin);
    if (counter > 0)
    {
        while (fread(&u, sizeof(u), 1, cstmr) > 0)
        {
            if ((strcmp(email, u.name) == 0) && (password == u.code))
            {
                fclose(cstmr);
                goto success;
                break;
            }
        }
        rewind(cstmr);
        counter--;
        system("cls");
        Beep(523, 500);
        colorprintf(3, "TRY AGAIN %d ATTEMPS REMAINING\n", counter);
        goto again;
    }
    else
    {
        Beep(523, 500);
        fclose(cstmr);
        colorprintf(0, "PLEASE CHECK YOUR EMAIL FOR NEW PASSWORD CODE\n");
        exit(1);
    }

success:
    colorprintf(0, "***WELCOME***\n");
    fclose(cstmr);
    delay(3);
    outlet();
}

```

```

void outlet()
{

    system("cls");
    order o1;
    items A;
    FILE *shop, *outlet;
    int i = 1, o_no = 0;
    float total;
    char numbers[12];
    o_no = orders_no();
    fflush(stdin);
    colorprintf(0, "\nPLEASE PROVIDE DELIVERY ADDRESS : "); // customer address
    gets(o1.c_address);
    fflush(stdin);
input_num:
    colorprintf(3, "\nPLEASE PROVIDE MOBILE NUMBER ACTIVE AT TIME OF DELIVERY : "
); //customer mobile no
    gets(numbers);
    fflush(stdin);
    if (strlen(numbers) != 11)
    {
        colorprintf(0, "\nPLEASE ENTER THE VALID MOBILE NUMBER\n");
        delay(2);
        system("cls");
        goto input_num;
    }
    for (int i = 0; i < strlen(numbers); i++)
    {
        if ((numbers[i] >= '0' && numbers[i] <= '9') );
        else
        {
            colorprintf(0, "\nPLEASE ENTER THE CORRECT MOBILE NUMBER\n");
            delay(2);
            system("cls");
            goto input_num;
        }
    }
    fflush(stdin);
    shop = fopen("shop.txt", "r");
more:

```

```

outlet = fopen("outlet.txt", "a");
rewind(shop);
if (shop == NULL)
{
    printf("error\n");
    exit(1);
}
system("cls");
//printing all data of shop file (all products)
printf("=====
=====\\n");
colorprintf(0, "S.NO\\tP.ID\\t\\t          P.NAME\\t\\t\\t          RS\\n");
printf("=====
=====\\n\\n");

while (fread(&A, sizeof(A), 1, shop) > 0)
{
    colorprintf(3, "%-3d\\t%-8d\\t\\t%-
20s\\t%8.3f\\n", i++, A.productid, A.productname, A.price);
}
i = 1;
printf("\\n=====
=====\\n\\n");

rewind(shop); //set file pointer to start

int buy, check = 0;
colorprintf(3, "ENTER THE PRODUCT CODE YOU WANT TO BUY : "); //to buy
scanf("%d", &buy);
while (fread(&A, sizeof(A), 1, shop) > 0)
{
    if (buy == A.productid) //if the given p_id is found then break the loop
    {
        colorprintf(0, "added to cart\\n\\n");
        check = 1;
        break;
    }
    // else printf("error!\\n");
}

rewind(shop);
if (check == 1) //rewind shop ptr
{
    total += A.price;                //total local variable (all product pr
ice added into total)

```



```

        o1.total_price = A.price;           //copy price to other struct
        o1.p_id = A.productid;             // copy
        strcpy(o1.product, A.productname); //copy
        o1.c_address;
        strcpy(o1.mobile_no, numbers);
        time(&o1.time);
        //o1.order_no=0;
        o1.order_no = o_no;
        // printf("                %d                ",o_no);
        srand(time(NULL));
        o1.ship_day = (rand() % 6) + 1;
        fwrite(&o1, sizeof(o1), 1, outlet);
        fclose(outlet);
    }
    else
    {
        Beep(523, 500);
        colorprintf(5, "INVALID PRODUCT ID\n\n");
    }

    int num;
again:
    colorprintf(0, "1  TO BUY MORE\n2  TO CHECK OUT\n3  TO VIEW CART\n");
    scanf("%d", &num);
    switch (num)
    {
        case 1:
            system("cls");
            goto more;
            break;
        case 2:
            system("cls");
            goto final;
            break;
        case 3:

            fclose(outlet);
            outlet = fopen("outlet.txt", "r");
            if (outlet == NULL)
            {
                printf("could not load \nerror");
                exit(1);
            }
    }

```

```

        printf("=====
=====
\n");
        colorprintf(0, "  PRICE\t\tPRODUCT\t\t      PRODUCT_ID \tSHIP_DAY  MOBIL
E_NO\t      ADDRESS\n");
        printf("=====
=====
\n\n");
        while (fread(&o1, sizeof(o1), 1, outlet) > 0)
        {
            if (o_no == o1.order_no)
            {
                colorprintf(3, "%-8.3f\t%-18s\b\t%-8d\t  %-2d\t  %-12s\t %-
5s", o1.total_price, o1.product, o1.p_id, o1.ship_day, o1.mobile_no, o1.c_address
);
            }
        }
        printf("\n=====
=====
==\n\n");
        fclose(outlet);
        int choose;
        colorprintf(0, "\n\npress 1 for checkout\npress 2 for buy more \n");
        scanf("%d", &choose);
        switch (choose)
        {
            case 1:
                system("cls");
                goto more;
                break;
            case 2:
                system("cls");
                goto final;
                break;
            default:
                Beep(523, 500);
                goto again;
                break;
        }
        default:
            Beep(523, 500);
            goto again;
        }
        fclose(outlet);
        fclose(shop);

```

```

final:
    colorprintf(0, "\n*****CHECK OUT*****\n");

    colorprintf(3, "\n\n***your total bill is  %f  PKR only***\n\n", total);
    fclose(outlet);
    fclose(shop);
    show_bill(o_no);
}

```

DELIVERY

➔ No one wants to sign up just to know delivery status so, you can go to delivery module and search you order by order id and mobile number.

```

void delivery()
{
    system("cls");
    colorprintf(0, "welcom to Quire about delivery\n\n");
    FILE *outlet;
    outlet = fopen("outlet.txt", "r");
    order A;
    if (outlet == NULL)
    {
        printf("error\n");
        exit(1);
    }
a:
    colorprintf(3, "search by\n1  MOBILE NUMBER\n2  BILL ID\n0  for exit\n");
    int num;
    scanf("%d", &num);
    fflush(stdin);
    switch (num)
    {
    case 1:
    {
        system("cls");

```

```

fflush(stdin);
char mobile[25];
colorprintf(0, "enter the mobile number\n");
//fgets(pname, sizeof(pname), stdin);
gets(mobile);
fflush(stdin);
printf("=====
=====
\n");
colorprintf(0, " PRICE\t\tPRODUCT\t\t PRODUCT_ID ORDER_NO\tSHIP_DAY
MOBILE_NO\t ADDRESS\t\t\t\t\tDATE\n");
printf("=====
=====
\n\n
");
while (fread(&A, sizeof(order), 1, outlet) > 0)
{
    if (strcmp(mobile, A.mobile_no) == 0)
    {
        colorprintf(3, "%-8.3f\t%-18s\b\t%-8d\t\b\b%-8d\t %-2d\t %-
12s\t %-
25s\t%s", A.total_price, A.product, A.p_id, A.order_no, A.ship_day, A.mobile_no,
A.c_address, ctime(&A.time));
    }

    //else printf("error!\n");
}
printf("=====
=====
\n\n");
break;
}
case 2:
{
    system("cls");
    fflush(stdin);
    int id;
    colorprintf(0, "enter the bill number\n");
    scanf("%d", &id);
    printf("=====
=====
\n\n
");
    colorprintf(0, " PRICE\t\tPRODUCT\t\t PRODUCT_ID ORDER_NO\tSHIP_DAY
MOBILE_NO\t ADDRESS\t\t\t\t\tDATE\n");
    printf("=====
=====
\n\n
");

```

```

        while (fread(&A, sizeof(A), 1, outlet) > 0)
        {
            if (id == A.order_no)
            {
                colorprintf(3, "%-8.3f\t%-18s\b\t%-8d\t\b\b%-8d\t    %-2d\t    %-
12s\t %-
25s\t%s", A.total_price, A.product, A.p_id, A.order_no, A.ship_day, A.mobile_no,
A.c_address, ctime(&A.time));
            }

            // else printf("error!\n");
        }
        printf("=====
=====\\n\\n");

        break;
    }
    case 0:
        colorprintf(0, "redirecting\\n");
        break;
    default:
        Beep(523, 500);
        colorprintf(0, "wrong input\\n");
        goto a;
    }
    colorprintf(0, "\\n****PRESS ANY KEY TO CONTINUE****\\n");

    getch();
    delay(2);
    fclose(outlet);
}

```

CUSTOMER CARE

```

void chatbot()
{
    system("cls");
    colorprintf(0, "****HI!! THIS IS ZEO, HOW MAY I HELP YOU :)****\\n \\ne.g (i want to know delivery status)\\n");

    char color[] = "color", deliverys[] = "deliver", Return[] = "return", zidello[] = "zidello", account[] = "account";
}

```

```

char input[35]; //if delivery is found then goto delivery function works
int count_color = 0, count_delivery = 0, count_Return = 0, count_zidello = 0,
count_account = 0;
start:
fflush(stdin);
colorprintf(0, "\nwhat you want to ask? : ");
gets(input);
for (int i = 0; i < strlen(input); i++)
{
    count_color = 0;
    count_delivery = 0;
    count_Return = 0;
    count_zidello = 0;
    count_account = 0;

    for (int j = 0; j < strlen(color); j++)
    {
        if (color[j] == input[j + i])
        {
            count_color++;
            if (count_color == strlen(color))
            {
                goto colors;
                break;
            }
        }
    }
    count_color = 0;
    for (int j = 0; j < strlen(account); j++)
    {
        if (account[j] == input[j + i])
        {
            count_account++;
            if (count_account == strlen(account))
            {
                goto accounts;
                break;
            }
        }
    }
    count_account = 0;
    for (int j = 0; j < strlen(deliverys); j++)
    {
        if (deliverys[j] == input[j + i])
        {

```

```

        count_delivery++;
        if (count_delivery == strlen(deliverys))
        {
            goto deliv;
            break;
        }
    }
}
count_delivery = 0;
for (int j = 0; j < strlen(Return); j++)
{
    if (Return[j] == input[j + i])
    {
        count_Return++;
        if (count_Return == strlen(Return))
        {
            goto ret;
            break;
        }
    }
}
count_Return = 0;
for (int j = 0; j < strlen(zidello); j++)
{
    if (zidello[j] == input[j + i])
    {
        count_zidello++;
        if (count_zidello == strlen(zidello))
        {
            goto zid;
            break;
        }
    }
}
count_zidello = 0;
}

int num;

colorprintf(3, "\nOH NO!,i can't understand please ask me what i can answer!\n");
fflush(stdin);
colorprintf(3, "\nif you want to ask any other thing press 1 : ");
scanf("%d", &num);
if (num == 1)

```

```

{
    system("cls");
    goto start;
}

colors:
    if (count_color == strlen(color))
        colorprintf(3, "\nWe work with RGB colors but if you want to know about a
ny specific product color \nthen call or message ZIDELLO's help center 0333-
1111111\n\n");

deliv:
    if (count_delivery == strlen(deliverys))
    {
        colorprintf(3, "\nWe deliver in working days\nPress D to know about your
delivery status\nPress any other key to continue\n");
        char choose;
        fflush(stdin);
        choose = getch();
        if (choose == 'D')
        {
            delay(3);
            delivery();
        }
    }

ret:
    if (count_Return == strlen(Return))
        colorprintf(3, "\nReturn policies are as follows \nIf you had removed the
seal then we will not accept.\nWe will not accept return, after a week of produc
t is bought.\nwe will not accept return if the product is broken or something else
.\nWe will not accept return if the the product is used.\n\n\nIf you are eligible
of returning a product then:\n1 Pack the product similarly as you get it.\n2 Ca
ll the ZIDELLO'S return center.\n3 Tell them about the credit how you want them
to return (vouchur/bank deposit).\n4 Wait for the pick-up man.\n ");

zid:
    if (count_zidello == strlen(zidello))
        colorprintf(3, "\nHI, How are you?\nZIDELLO's is the flourishing e-
shopping website with the othentic sellors\nour aim is to serve as many features
as we can to our customers.\n");

accounts:
    if (count_account == strlen(account))
        colorprintf(3, "\nNEW USER?\ngoto customer portal and signup\n\nForget pa
ssword?\ncall on Zidello's help center 0333-

```



```

1111111\n or if you are facing any other problem e.g(hacked) contact Zidello's help center 0333-1111111\n");

    colorprintf(0, "\n****Thank you have a nice day****\n\nFor more please call ZIDELLO's help center 0333-1111111\n\nPress any key to continue\n");
    getch();
    delay(2);
}

//    BY SHARJEEL AHMED

```

OTHER ASSISTING FUNCTIONS

```

int product_id() //to assign unique order no to every product
{
    system("cls");
    items A;
    FILE *shop;
    shop = fopen("shop.txt", "r");
    fseek(shop, 0, SEEK_END);
    if (0 == ftell(shop)) //if file is empty return 10000
    {
        rewind(shop);
        fclose(shop);
        int p_id = 10000;
        return p_id += 1;
    }
    rewind(shop);
    while (fread(&A, sizeof(A), 1, shop) > 0)
        ;

    fclose(shop);
    return A.productid + 1; //if file is not empty return last p_id+1
}

```

➔ This function assigns unique product id to every product

```

int orders_no() // to assign unique order no
{
    system("cls");
    order o;
    FILE *outlet;
    outlet = fopen("outlet.txt", "r");
    if(outlet==NULL)
    {
        return 1001;
    }
    fseek(outlet, 0, SEEK_END);
    if (0 == ftell(outlet)) //if file is empty return 1000
    {
        rewind(outlet);
        fclose(outlet);
        return 1234;
    }
    rewind(outlet);
    while (fread(&o, sizeof(o), 1, outlet) > 0)
        ;

    fclose(outlet);
    return o.order_no + 1; //if file has data return last order_no+1
}

```

➔ This function assigns unique product id to every generated bill

```

void show_bill(int code) //to generate bill of code which is provided by the bill
generator (comming from outlet)
{
    FILE *outlet;
    order show;
    outlet = fopen("outlet.txt", "r");
    if (outlet == NULL)
    {
        printf("could not load \nerror");
        exit(1);
    }
}

```

```

        printf("=====
=====\\n")
;
    colorprintf(0, "  PRICE\\t\\tPRODUCT\\t\\t      PRODUCT_ID      ORDER_NO\\tSHIP_DAY
MOBILE_NO\\t      ADDRESS\\t\\t\\t\\tDATE\\n");
    printf("=====
=====\\n\\n
");
    while (fread(&show, sizeof(show), 1, outlet) > 0)
    {
        if (code == show.order_no)
        {
            colorprintf(3, "%-8.3f\\t%-18s\\b\\t%-8d\\t\\b\\b%-8d\\t  %-2d\\t  %-
12s\\t %-
25s\\t%s", show.total_price, show.product, show.p_id, show.order_no, show.ship_day
, show.mobile_no, show.c_address, ctime(&show.time));
        }
    }
    printf("\\n=====
=====\\n
\\n");
    fclose(outlet);
    colorprintf(0, "press any key to continue\\n");
    delay(1);
    getch();
}

```

➔ To generate total bill of customer

```

void H() // just for printing H
{
    int height = 7;
    int width = (2 * height) - 1;

    int i, j, k = 0;
    for (i = 0; i < height; i++)
    {
        delayprint(2);
        colorprintf(k++, "  *");
        if (k == 5)
            k = 0;
    }
}

```

```

    for (j = 0; j < height; j++)
    {
        if ((j == height - 1) || (i == height / 2))
        {
            delayprint(2);
            colorprintf(k++, "*");
            if (k == 5)
                k = 0;
        }
        else
        {
            delayprint(0); // delay function call
            printf(" ");
        }
    }

    printf("\n");
}

```

➔ Pattern of 'H'

```

void I() //just for printing I
{
    int height = 7, i, j, k = 0;
    int width = (2 * height) - 1;
    for (i = 0; i < height; i++)
    {
        for (j = 0; j < height; j++)
        {
            if (i == 0 || i == height - 1)
            {
                delayprint(2); //delay every time it print
                colorprintf(k++, "*");
                if (k == 5)
                    k = 0;
            }
            else if (j == height / 2)
            {
                delayprint(2);
                colorprintf(k++, "*");
                if (k == 5)

```

```

        k = 0;
    }
    else
    {
        delayprint(0);
        printf(" ");
    }
}
printf("\n");
}
}

```

➔ Pattern of 'I'

```

void delayprint(int no_of_sec)
{
    clock_t start = clock();
    int i = 0;
    for (i = 0; i < no_of_sec; i++)
    {
        while (clock() < start + (i * 600))
            ;
    }
}

```

➔ For printing every character after 0.6 sec

```

void delay(int no_of_sec)
{
    clock_t start = clock(); // start time
    int i = 0, j = 0;
    for (i = 0; i < no_of_sec; i++)
    {
        while (clock() < start + (i * 900)) //0.9 sec
            ;
        colorprintf(j, ".");
        if (j == 5)
            j = 0;
    }
}

```

```
j++;  
}  
}
```

➔ For printing dots ‘.’

NOTE: Colorprintf is a library which is imported from chrome.