CLOTHING MANAGEMENT SYSTEM

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

NAVANEETH KRISHNA TANGUTURI – 1602-19-737-138

SATWIK KASUKURTHI – 1602-19-737-165

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)

2020

Hyderabad-500 031

Department of Information Technology

DECLARATION BY THE CANDIDATE

We NAVANEETH KRISHNA TANGUTURI and SATWIK KASUKURTHI bearing hall ticket numbers, 1602-19-737-138 and 1602-19-737-165 respectively, hereby declare that the project report entitled "CLOTHING MANAGEMENT SYSTEM" 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 Bonafede work carried out by us 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.

NAVANEETH KRISHNA
TANGUTURI
1602-19-737-138

SATWIK KASUKURTHI
1602-19-737-165

(Faculty In- Charge) (Head, Dept. of IT)

AKNOWLEDGEMENTS

We wouldn’t have completed our Mini Project without the help of several people. We are extremely thankful to our college, Vasavi College of Engineering, for providing the opportunity to implement our project, “CLOTHING MANAGEMENT SYSTEM”.

We are also extremely thankful to Ms. Divya Lingineni, Assistant Professor, Department of Information Technology, Vasavi College of Engineering and Dr. Ramesh Vasappanavara, Professor, Department of Information Technology, Vasavi College of Engineering, for their esteemed guidance, moral support and invaluable advice provided by them for the success of the Mini Project.

Sincerely,

NAVANEETH KRISHNA TANGUTURI 1602-19-737-138

SATWIK KASUKURTHI 1602-19-737-165

ABSTRACT

Today's world demands only speed and efficiency. The existing clothing management is done on paper. This requires a lot of manual work, resources, capital and is more error prone to making errors. "CLOTHING MANAGEMENT SYSTEM" is a Mini Project with the hope of replacing the traditional way of maintaining clothing related information to increase efficiency and reduce manual work, resources and capital. We hope to fully computerize the existing system which will be easy to use and efficient overall.

TABLE OF CONTENTS

1. INTRODUCTION

1.1. ABOUT THE PROJECT

“CLOTHING MANAGEMENT SYSTEM” is a Mini Project with the goal to replace existing manual system to store information whether it may be on paper or any other form of hardcopy to increase speed and efficiency. It is a console-based application which is written is C language

1.2. PROJECT DOMAIN

The existing clothing management is done on paper. This requires a lot of manual work, resources, capital and is more error prone to making errors. The existing records are also very easy to access by other people This problem can be tackled with the help of our console-based Clothing Management System which provides an interface for the user which is easy to use. The entire interface is password protected to add extra layer of protection. Our project is also very efficient in storing and organising the clothing related information and is easier to edit, update and maintain as opposed to more traditional ways of entering clothing related information. The project falls under Console Application which is a text-only interface. It displays features which the user can interact with making the total experience smooth and hassle free.

1.3. FEATURES

“CLOTHING MANAGEMENT SYSTEM” is a two ended console-based application. It has **user** and **admin** at either side of the end. Before all a user must already be registered in the file handled database in order to access the user features.

1.3.1. LOGIN

Right at the beginning the console will display three options as to allow the one behind the monitor to login as an admin or a user or if hes not a registered as a user he could register as one. If the person is a registered user, the console will prompt the person to enter his user name and password. If the login credentials are correct; the user will be taken to the user page where they can access user features.

If the person behind the monitor is an admin, the console will prompt the person to enter the user name and password. The username and password of an admin are hard coded into code. If the credentials match, the admin will be taken to the admin page where they can access admin features.

If the person behind the monitor is a new user, the console will prompt the person to enter a username. Then it checks the database whether the username is not taken by a pre-existing user. If the username is unique, they are to enter a password and then they are taken to login page where they can login or exit the console.

1.3.2. USER FEATURES

User features include

- Viewing all items in the stores inventory
- Searching for an already existing item
- Selecting existing items
- Billing the selected items or cancel the whole checkout

1.3.3. ADMIN FEATURES

Admin features include

- Editing details of an existing item
- Viewing all the items in the stores inventory
- Adding a new item into the stores inventory
- Deleting an existing item
- Searching for an already existing item

2. TECHNOLOGY

"CLOTHING MANAGEMENT SYSTEM" is a very lite application and would not require very high amounts of system resources.

All computer software needs certain hardware components or other software resources to be present, in order for computers to be used efficiently. These prerequisites are known as System Requirements. Within this, we have two types – Software Requirements and Hardware Requirements.

2.1. SOFTWARE REQUIREMENTS

Software Requirements deal with defining the software resource requirements and prerequisites that need to be installed on a computer to provide optimal functioning of an application. These preconditions are generally not included in the software installation package and need to be installed separately.

- **Operating System:** Windows 7 and above
- **C Compiler:** GNU Compiler Collection (GCC)
- **Editor:** Any text editor which can be used to read C Language code. (Note: The project was build using Dev C++ Editor).

2.2. HARDWARE REQUIREMENTS

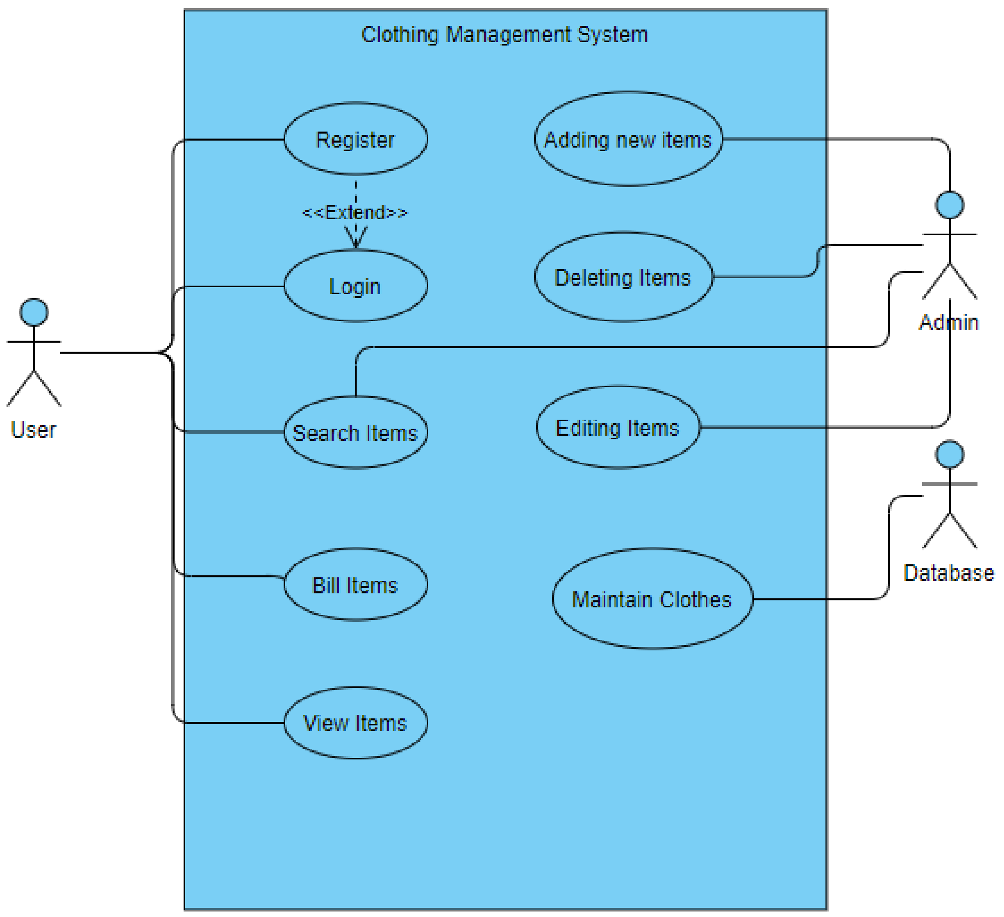
Hardware requirements refer to the common set requirements defined by any operating system or software application and are usually the physical computer resources. In this, we look into the architecture, processing power, memory, secondary memory, display adapter and peripherals.

- **Processor:** Intel Core i5 and above
- **Memory:** 8 GB RAM

3. PROPOSED WORK

3.1. DESIGN

The users of “CLOTHING MANAGEMENT SYSTEM” are divided into two categories namely **Users** and **Admin**. A **user** is the one who comes to the *store* to **view** all the items present in the stores inventory or they could **search** for a certain already existing item in the inventory or they could **select** a bunch of items and could procced to **billing** or could exit the terminal. An **admin** is a the one with special access who could **add** new items into the inventory, **edit** the clothing related details or **delete** already existing items from the inventory. Along with those the admin could **view** all items present in the inventory and **search** for an already existing item in the inventory.



User Case Diagram

3.1.1. USER USE CASES

User has 6 functionalities: Register, Login, Search item, Select Items, View Items and Bill Items.

3.1.1.1. REGISTER

If the person behind the monitor is a new user, the console will prompt the person to enter a username. Then it checks the database whether the username is not taken by a pre-existing user. If the username is unique, they are to enter a password and then they are taken to login page where they can login. Upon successfully logging in, the user can access user features.

3.1.1.2. LOGIN

If the person is a registered user, the console will prompt the person to enter his user name and password. If the login credentials are correct; the user will be taken to the user page where they can access user features.

3.1.1.3. SEARCH ITEM

Once the user has successfully logged in, the console will list the user features one of which is Search Item. Upon selection of Search Item, the console will prompt the user to enter a keyword. A keyword could be colour, brand, price, item name. The keyword is searched in the database and upon matching, it displays the entire item details on the console.

3.1.1.4. SELECT ITEMS

Once the user has successfully logged in, the console will list the user features one of which is Select Item. Upon selection of Select Item, the console will prompt the user to enter the item ID of the selected item and the quantity of requirement. After selecting the items that the user selected, the console will ask the user whether they want to head towards the billing page. If yes, the console will take the user to the billing page else the console is terminated.

3.1.1.5. VIEW ITEMS

Once the user has successfully logged in, the console will list the user features one of which is View Items. Upon selection it prints the Item Details such as Item Name, Item Price, Item Brand, Item Type, Item Colour.

3.1.1.6. BILL ITEMS

After selecting items and upon heading towards billing in the select item page, the user is brought to the bill items page where the selected items and the quantity are checked and finally the total bill is displayed

3.1.2. ADMIN USE CASES

Admin has 6 functionalities. Login, Adding New Items, Editing Items, Deleting Items, Search Items and View Items.

3.1.2.1. LOGIN

If the person behind the monitor is an admin, the console will prompt the person to enter the user name and password. The username and password of an admin are hard coded into code. If the credentials match, the admin will be taken to the admin page where they can access admin features

3.1.2.2. ADDING NEW ITEMS

Once the admin has successfully logged in, the console will list admin features out of which one is Adding New Items. Upon selection the console will ask the admin to enter the new item’s Item ID. Then the Item ID is checked in the database and if it exists, it prompts the user to enter a unique Item ID. After a unique Item ID is entered, the console prompts the user enter further details which are Item Name, Item Price, Item Colour, ItemType and Item Brand

.

3.1.2.3. EDITING ITEMS

Once the admin has successfully logged in, the console will list admin features out of which one is Editing Item. Upon selection the console prompt the user to enter the Item ID of the item whose details are to be edited. Then the console will prompt the admin which one of Item Name, Item Price, Item Brand, Item Colour or Item Type is to be edited. The admin can choose one of those five and the console will prompt the admin to enter new data and the change will be reflected in the database.

3.1.2.4. DELETING ITEMS

Once the admin has successfully logged in, the console will list admin features out of which one deleting item. The console will prompt the admin to enter the Item ID of the item which is to be deleted and will check if it is present in the database. Upon match the item will be deleted successfully.

3.1.2.5. SEARCH ITEM

Once the admin has successfully logged in, the console will list admin features out of which one is Search Item. Upon selection of Search Item, the console will prompt the user to enter a keyword. A keyword could be colour, brand, price, item name. The keyword is searched in the database and upon matching, it displays the entire item details on the console.

3.1.2.6. VIEW ITEMS

Once the admin has successfully logged in, the console will list admin features out of which one is View Item. Upon selection it prints the Item Details such as Item Name, Item Price, Item Brand, Item Type, Item Colour

3.2. IMPLEMENTATION

3.2.1. MODULE-WISE CODE

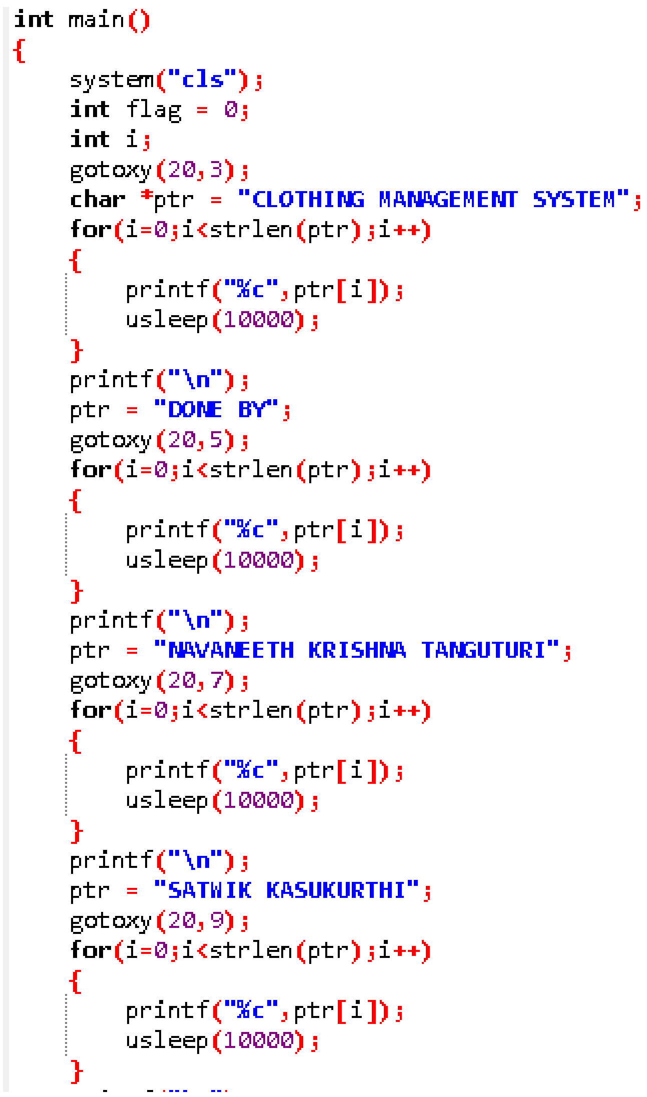
The data in the text files are accessed by structures.

```
struct User
{
    char username[50];
    char password[10];
}user;

struct Item
{
    char itemID[10];
    char itemName[10];
    char itemPrice[10];
    char colour[10];
    char itemType[10];
    char itemBrand[10];
}*item;
```

gotoxy ()

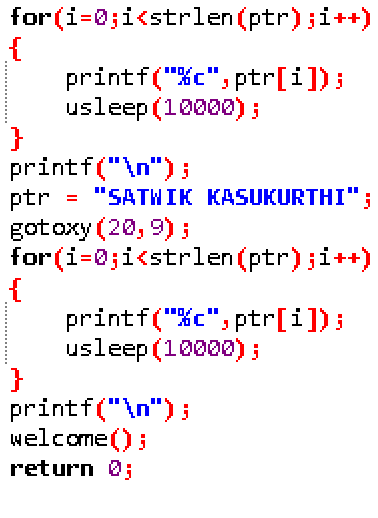
```
void gotoxy(int x, int y)
{
    COORD coord;
    coord.X = x;
    coord.Y = y;
    SetConsoleCursorPosition(GetStdHandle(STD_OUTPUT_HANDLE), coord);
}
```

main ()



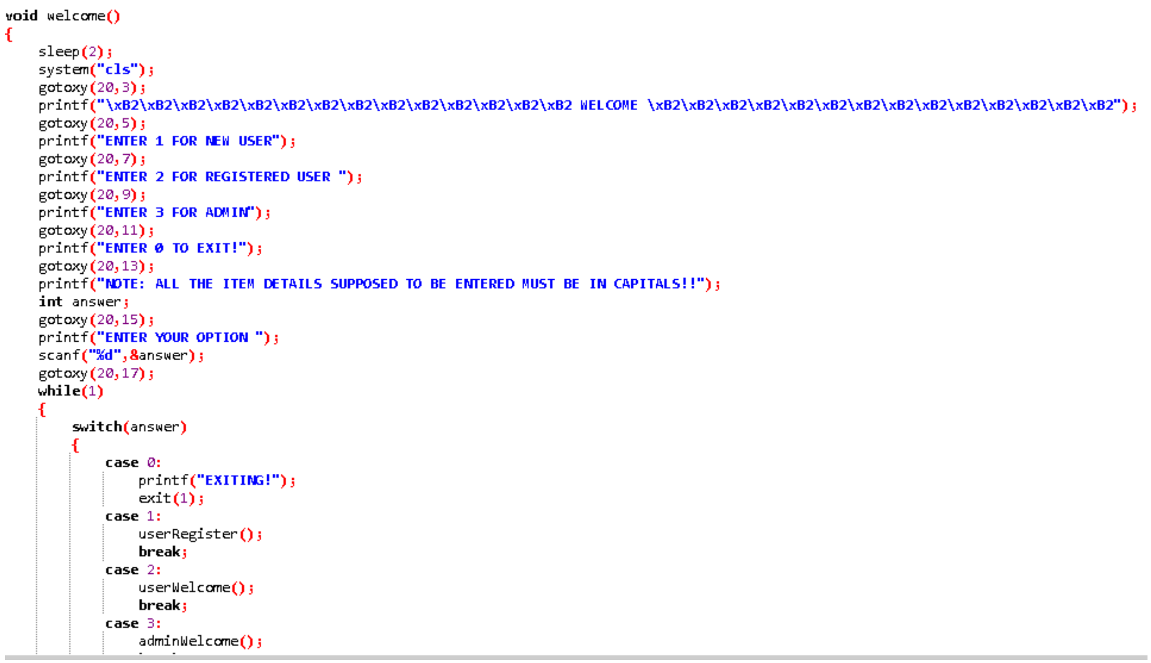
```c
int main()
{
    system("cls");
    int flag = 0;
    int i;
    gotoxy(20,3);
    char *ptr = "CLOTHING MANAGEMENT SYSTEM";
    for(i=0;i<strlen(ptr);i++)
    {
        printf("%c",ptr[i]);
        usleep(10000);
    }
    printf("\n");
    ptr = "DONE BY";
    gotoxy(20,5);
    for(i=0;i<strlen(ptr);i++)
    {
        printf("%c",ptr[i]);
        usleep(10000);
    }
    printf("\n");
    ptr = "NAVANEETH KRISHNA TANGUTURI";
    gotoxy(20,7);
    for(i=0;i<strlen(ptr);i++)
    {
        printf("%c",ptr[i]);
        usleep(10000);
    }
    printf("\n");
    ptr = "SATWIK KASUKURTHI";
    gotoxy(20,9);
    for(i=0;i<strlen(ptr);i++)
    {
        printf("%c",ptr[i]);
        usleep(10000);
    }
```

....

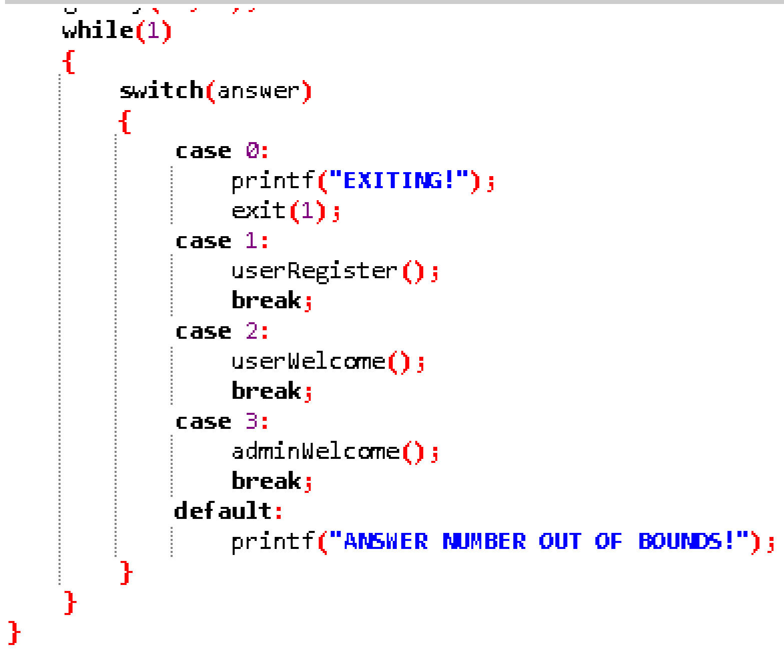


```c
    for(i=0;i<strlen(ptr);i++)
    {
        printf("%c",ptr[i]);
        usleep(10000);
    }
    printf("\n");
    ptr = "SATWIK KASUKURTHI";
    gotoxy(20,9);
    for(i=0;i<strlen(ptr);i++)
    {
        printf("%c",ptr[i]);
        usleep(10000);
    }
    printf("\n");
    welcome();
    return 0;
}
```

welcome ()

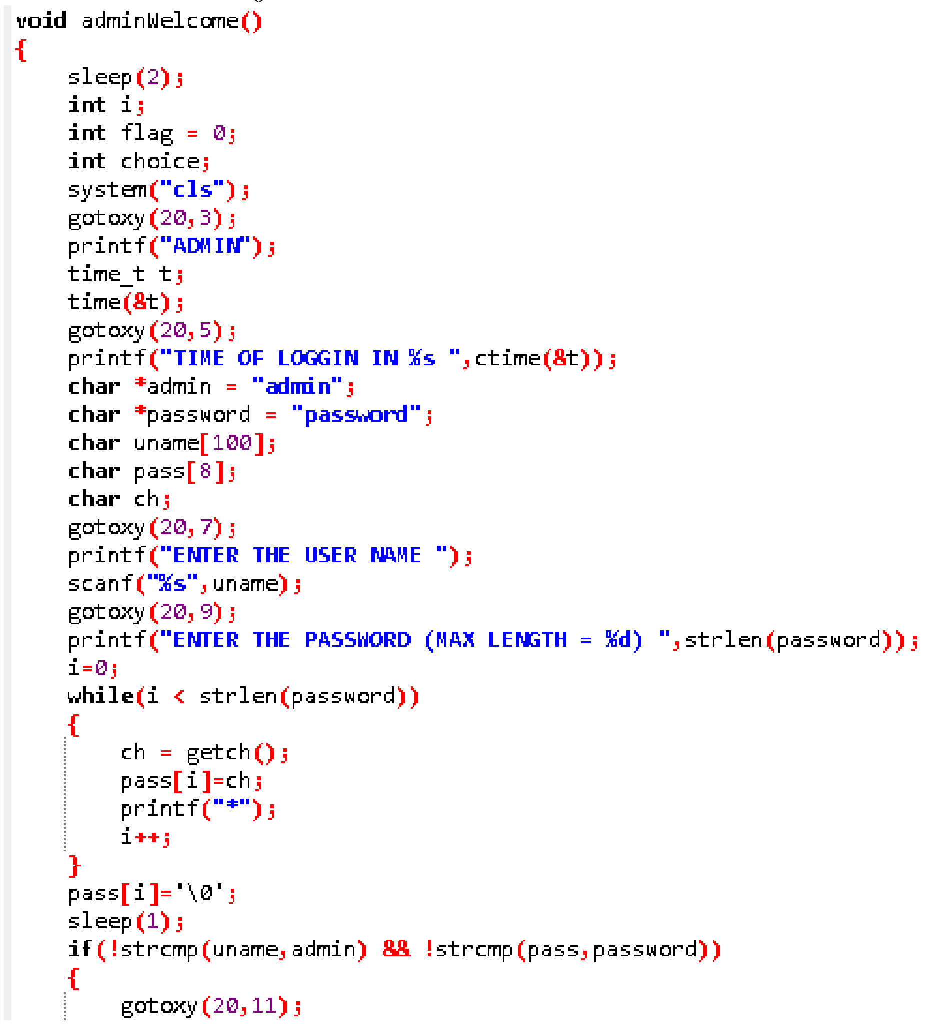


```
void welcome()
{
    sleep(2);
    system("cls");
    gotoxy(20,3);
    printf("\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2 WELCOME \xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2");
    gotoxy(20,5);
    printf("ENTER 1 FOR NEW USER");
    gotoxy(20,7);
    printf("ENTER 2 FOR REGISTERED USER ");
    gotoxy(20,9);
    printf("ENTER 3 FOR ADMIN");
    gotoxy(20,11);
    printf("ENTER 0 TO EXIT!");
    gotoxy(20,13);
    printf("NOTE: ALL THE ITEM DETAILS SUPPOSED TO BE ENTERED MUST BE IN CAPITALS!!");
    int answer;
    gotoxy(20,15);
    printf("ENTER YOUR OPTION ");
    scanf("%d",&answer);
    gotoxy(20,17);
    while(1)
    {
        switch(answer)
        {
            case 0:
                printf("EXITING!");
                exit(1);
            case 1:
                userRegister();
                break;
            case 2:
                userWelcome();
                break;
            case 3:
                adminWelcome();
```

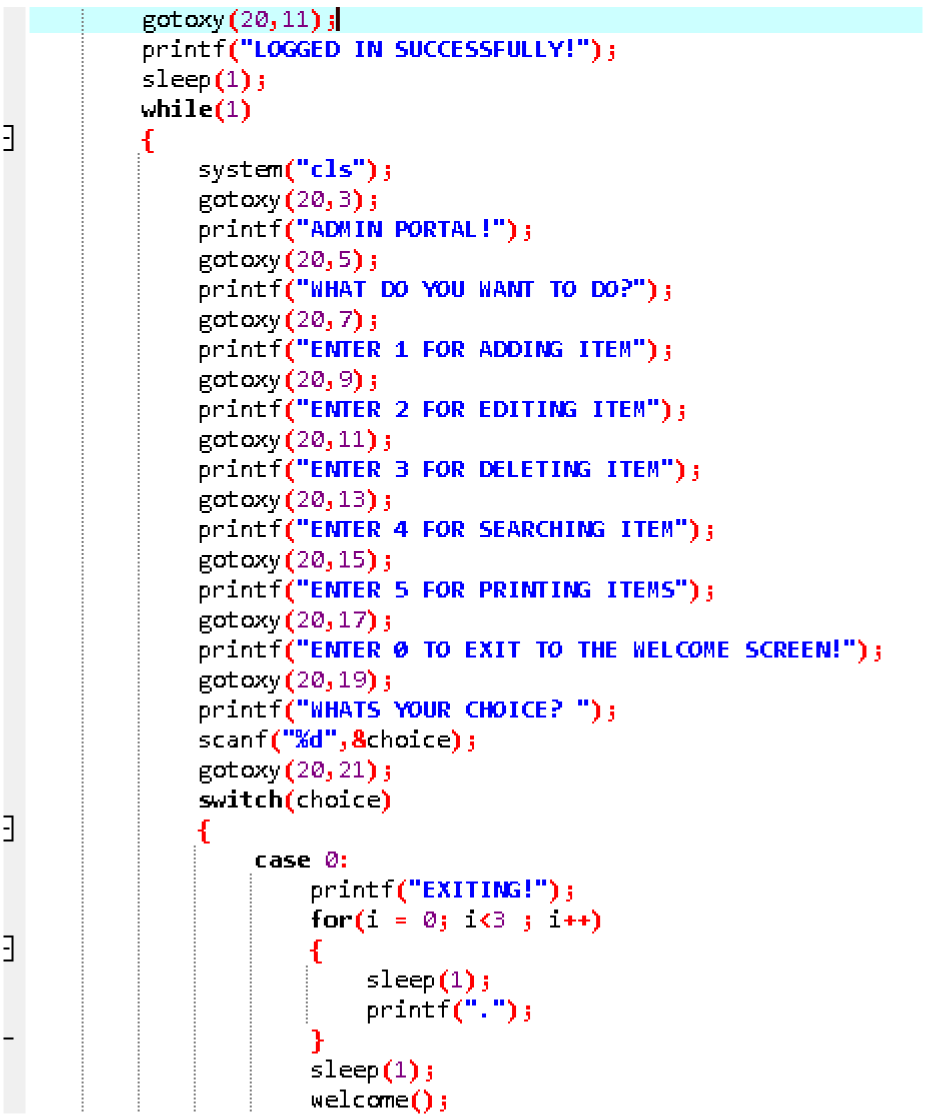


```
    while(1)
    {
        switch(answer)
        {
            case 0:
                printf("EXITING!");
                exit(1);
            case 1:
                userRegister();
                break;
            case 2:
                userWelcome();
                break;
            case 3:
                adminWelcome();
                break;
            default:
                printf("ANSWER NUMBER OUT OF BOUNDS!");
        }
    }
}
```

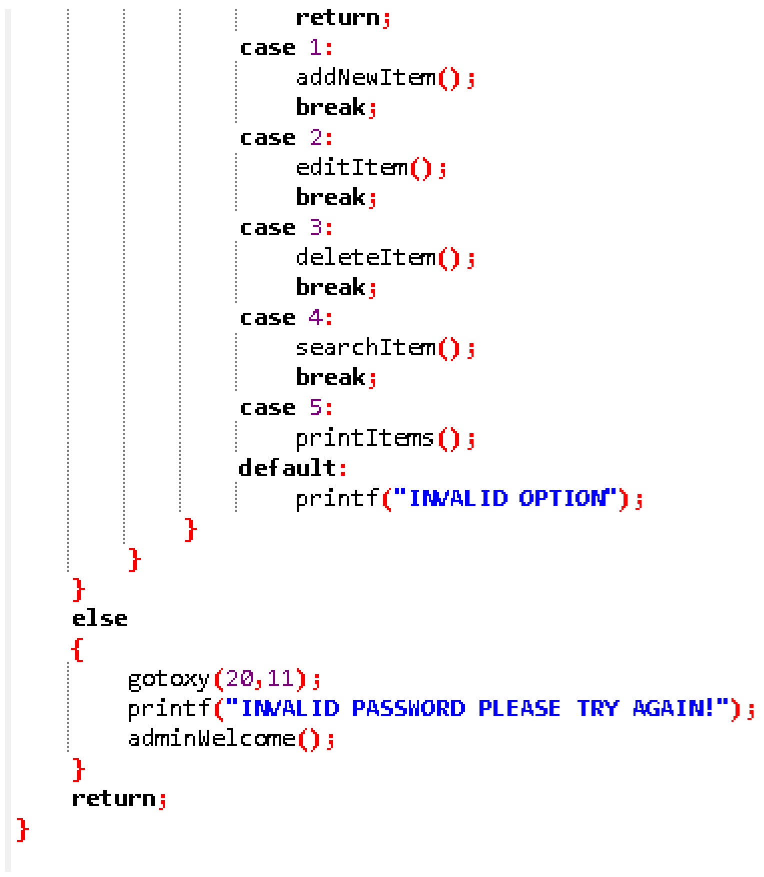
adminWelcome()



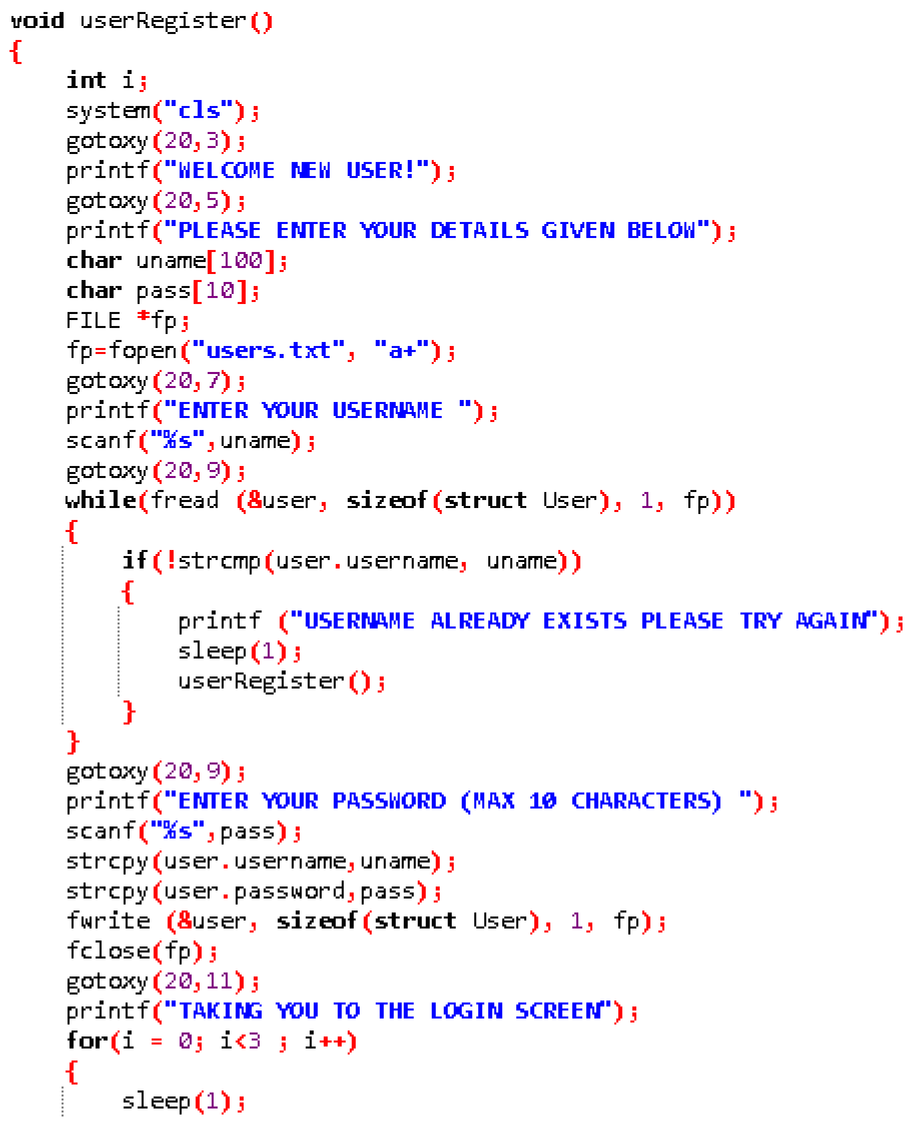
```
void adminWelcome()
{
    sleep(2);
    int i;
    int flag = 0;
    int choice;
    system("cls");
    gotoxy(20,3);
    printf("ADMIN");
    time_t t;
    time(&t);
    gotoxy(20,5);
    printf("TIME OF LOGGIN IN %s ",ctime(&t));
    char *admin = "admin";
    char *password = "password";
    char uname[100];
    char pass[8];
    char ch;
    gotoxy(20,7);
    printf("ENTER THE USER NAME ");
    scanf("%s",uname);
    gotoxy(20,9);
    printf("ENTER THE PASSWORD (MAX LENGTH = %d) ",strlen(password));
    i=0;
    while(i < strlen(password))
    {
        ch = getch();
        pass[i]=ch;
        printf("*");
        i++;
    }
    pass[i]='\0';
    sleep(1);
    if(!strcmp(uname,admin) && !strcmp(pass,password))
    {
        gotoxy(20,11);
```



```c
        gotoxy(20,11);
        printf("LOGGED IN SUCCESSFULLY!");
        sleep(1);
        while(1)
        {
            system("cls");
            gotoxy(20,3);
            printf("ADMIN PORTAL!");
            gotoxy(20,5);
            printf("WHAT DO YOU WANT TO DO?");
            gotoxy(20,7);
            printf("ENTER 1 FOR ADDING ITEM");
            gotoxy(20,9);
            printf("ENTER 2 FOR EDITING ITEM");
            gotoxy(20,11);
            printf("ENTER 3 FOR DELETING ITEM");
            gotoxy(20,13);
            printf("ENTER 4 FOR SEARCHING ITEM");
            gotoxy(20,15);
            printf("ENTER 5 FOR PRINTING ITEMS");
            gotoxy(20,17);
            printf("ENTER 0 TO EXIT TO THE WELCOME SCREEN!");
            gotoxy(20,19);
            printf("WHATS YOUR CHOICE? ");
            scanf("%d",&choice);
            gotoxy(20,21);
            switch(choice)
            {
                case 0:
                    printf("EXITING!");
                    for(i = 0; i<3 ; i++)
                    {
                        sleep(1);
                        printf(".");
                    }
                    sleep(1);
                    welcome();
```

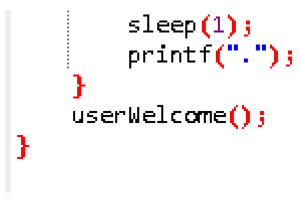

```
                    return;
                case 1:
                    addNewItem();
                    break;
                case 2:
                    editItem();
                    break;
                case 3:
                    deleteItem();
                    break;
                case 4:
                    searchItem();
                    break;
                case 5:
                    printItems();
                default:
                    printf("INVALID OPTION");
            }
        }
    }
    else
    {
        gotoxy(20,11);
        printf("INVALID PASSWORD PLEASE TRY AGAIN!");
        adminWelcome();
    }
    return;
}
```

userRegister()



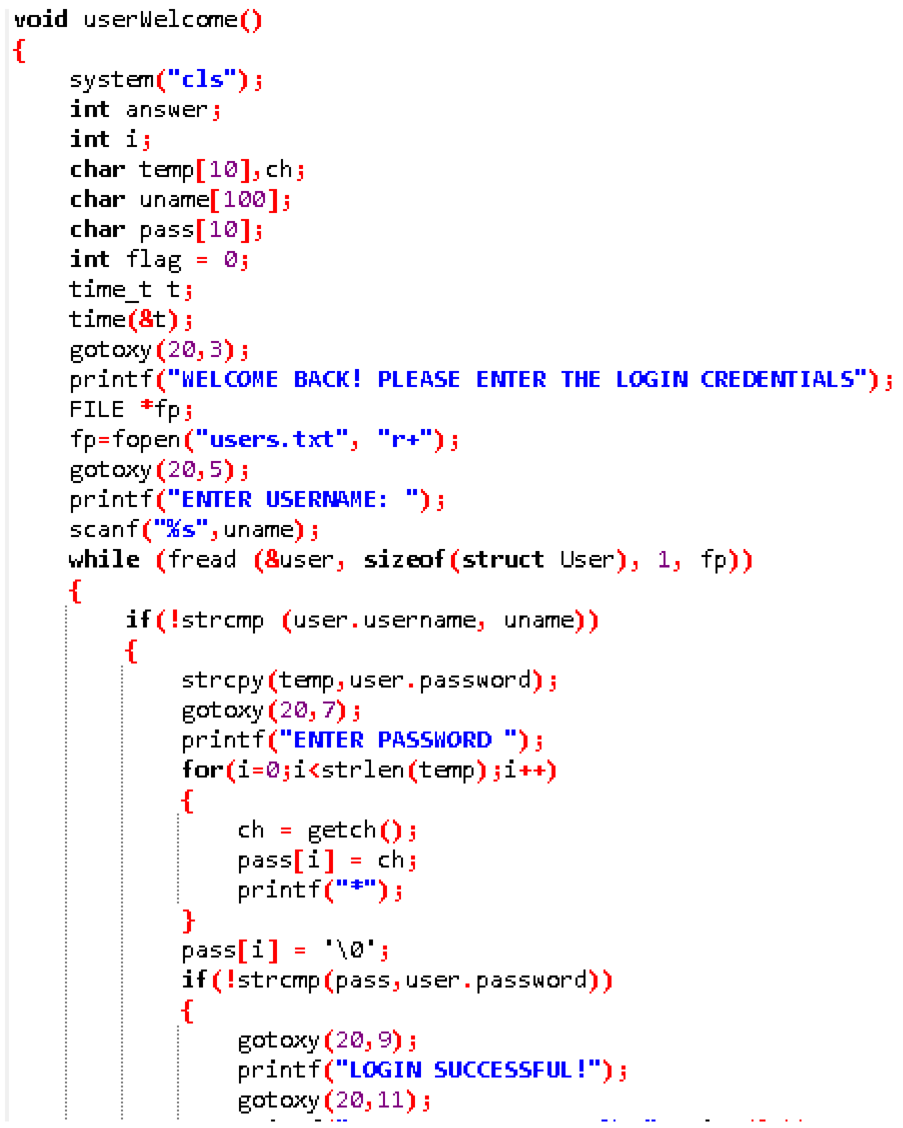
```c
void userRegister()
{
    int i;
    system("cls");
    gotoxy(20,3);
    printf("WELCOME NEW USER!");
    gotoxy(20,5);
    printf("PLEASE ENTER YOUR DETAILS GIVEN BELOW");
    char uname[100];
    char pass[10];
    FILE *fp;
    fp=fopen("users.txt", "a+");
    gotoxy(20,7);
    printf("ENTER YOUR USERNAME ");
    scanf("%s",uname);
    gotoxy(20,9);
    while(fread (&user, sizeof(struct User), 1, fp))
    {
        if(!strcmp(user.username, uname))
        {
            printf ("USERNAME ALREADY EXISTS PLEASE TRY AGAIN");
            sleep(1);
            userRegister();
        }
    }
    gotoxy(20,9);
    printf("ENTER YOUR PASSWORD (MAX 10 CHARACTERS) ");
    scanf("%s",pass);
    strcpy(user.username,uname);
    strcpy(user.password,pass);
    fwrite (&user, sizeof(struct User), 1, fp);
    fclose(fp);
    gotoxy(20,11);
    printf("TAKING YOU TO THE LOGIN SCREEN");
    for(i = 0; i<3 ; i++)
    {
        sleep(1);
```

….

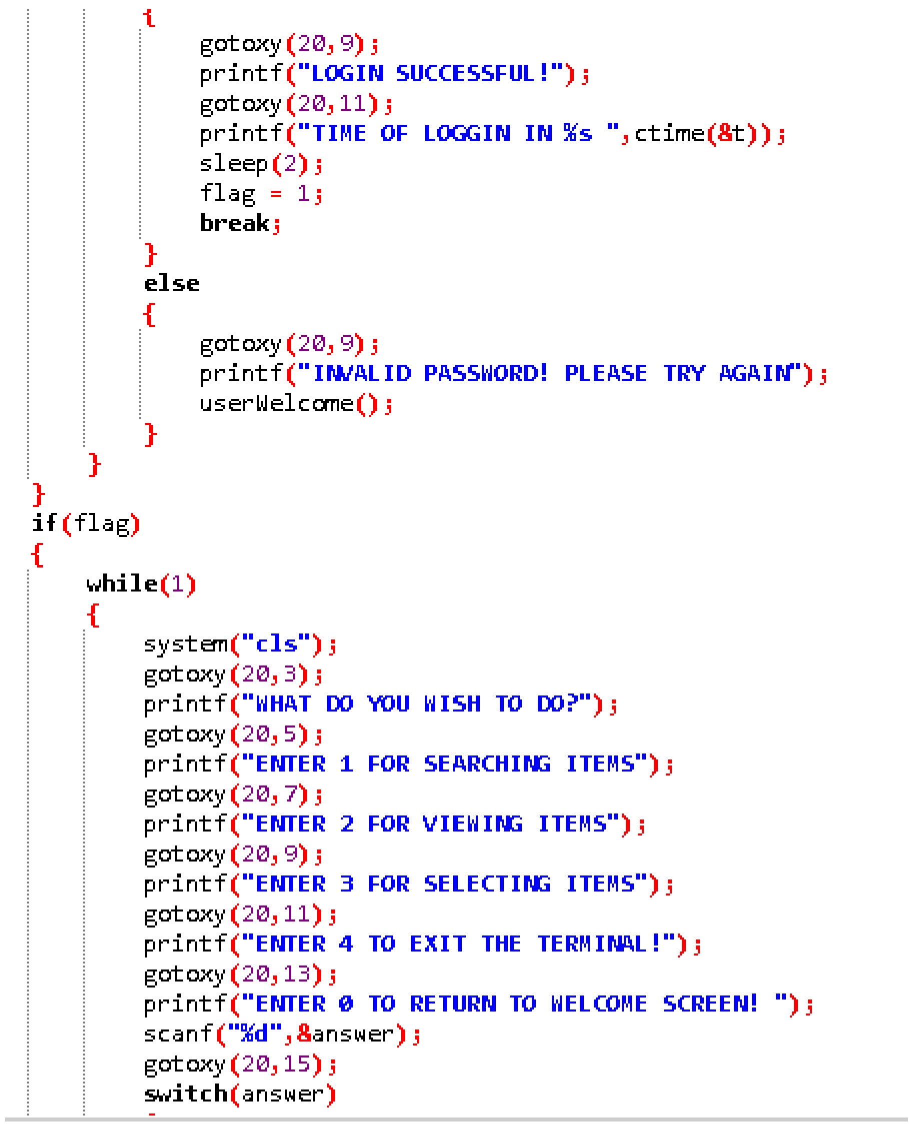


```c
        sleep(1);
        printf(".");
    }
    userWelcome();
}
```

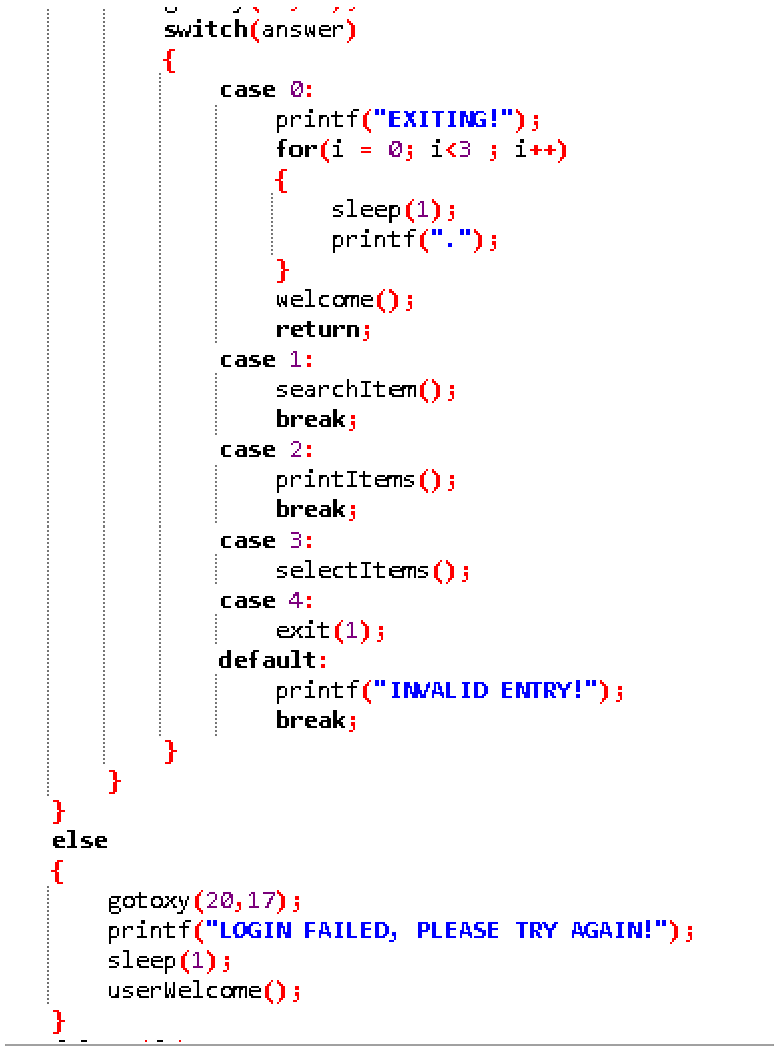
userWelcome()



```c
void userWelcome()
{
    system("cls");
    int answer;
    int i;
    char temp[10],ch;
    char uname[100];
    char pass[10];
    int flag = 0;
    time_t t;
    time(&t);
    gotoxy(20,3);
    printf("WELCOME BACK! PLEASE ENTER THE LOGIN CREDENTIALS");
    FILE *fp;
    fp=fopen("users.txt", "r+");
    gotoxy(20,5);
    printf("ENTER USERNAME: ");
    scanf("%s",uname);
    while (fread (&user, sizeof(struct User), 1, fp))
    {
        if(!strcmp (user.username, uname))
        {
            strcpy(temp,user.password);
            gotoxy(20,7);
            printf("ENTER PASSWORD ");
            for(i=0;i<strlen(temp);i++)
            {
                ch = getch();
                pass[i] = ch;
                printf("*");
            }
            pass[i] = '\0';
            if(!strcmp(pass,user.password))
            {
                gotoxy(20,9);
                printf("LOGIN SUCCESSFUL!");
                gotoxy(20,11);
```

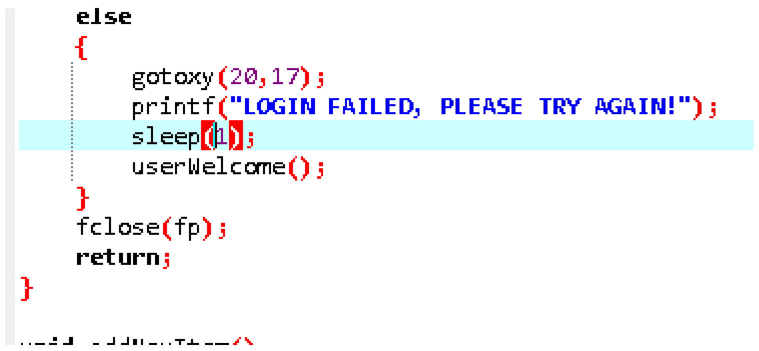


```
        {
            gotoxy(20,9);
            printf("LOGIN SUCCESSFUL!");
            gotoxy(20,11);
            printf("TIME OF LOGGIN IN %s ",ctime(&t));
            sleep(2);
            flag = 1;
            break;
        }
        else
        {
            gotoxy(20,9);
            printf("INVALID PASSWORD! PLEASE TRY AGAIN");
            userWelcome();
        }
    }
}
if(flag)
{
    while(1)
    {
        system("cls");
        gotoxy(20,3);
        printf("WHAT DO YOU WISH TO DO?");
        gotoxy(20,5);
        printf("ENTER 1 FOR SEARCHING ITEMS");
        gotoxy(20,7);
        printf("ENTER 2 FOR VIEWING ITEMS");
        gotoxy(20,9);
        printf("ENTER 3 FOR SELECTING ITEMS");
        gotoxy(20,11);
        printf("ENTER 4 TO EXIT THE TERMINAL!");
        gotoxy(20,13);
        printf("ENTER 0 TO RETURN TO WELCOME SCREEN! ");
        scanf("%d",&answer);
        gotoxy(20,15);
        switch(answer)
```



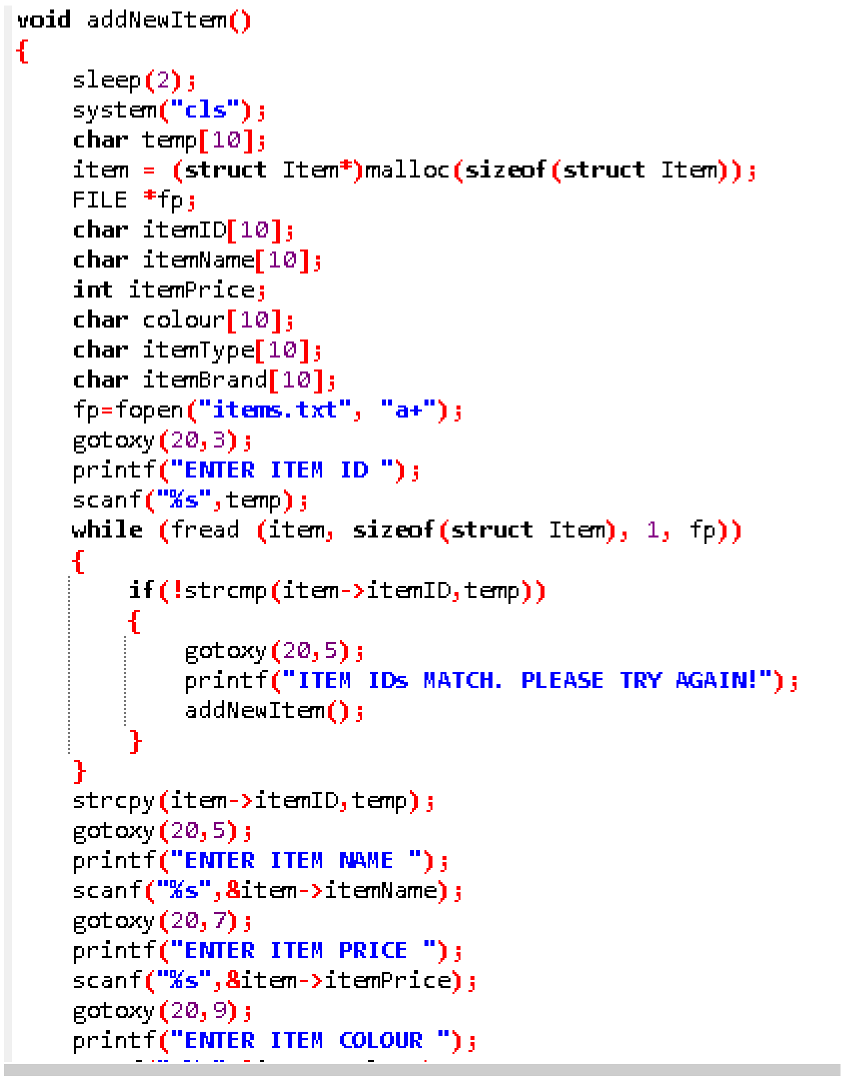
```
        switch(answer)
        {
            case 0:
                printf("EXITING!");
                for(i = 0; i<3 ; i++)
                {
                    sleep(1);
                    printf(".");
                }
                welcome();
                return;
            case 1:
                searchItem();
                break;
            case 2:
                printItems();
                break;
            case 3:
                selectItems();
            case 4:
                exit(1);
            default:
                printf("INVALID ENTRY!");
                break;
        }
    }
}
else
{
    gotoxy(20,17);
    printf("LOGIN FAILED, PLEASE TRY AGAIN!");
    sleep(1);
    userWelcome();
}
```

•



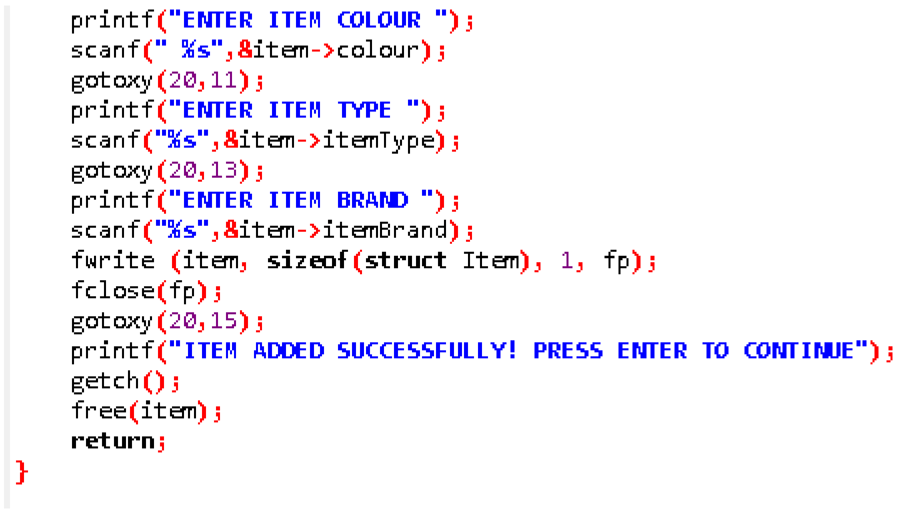
```
    else
    {
        gotoxy(20,17);
        printf("LOGIN FAILED, PLEASE TRY AGAIN!");
        sleep(1);
        userWelcome();
    }
    fclose(fp);
    return;
}
```

addNewItem()



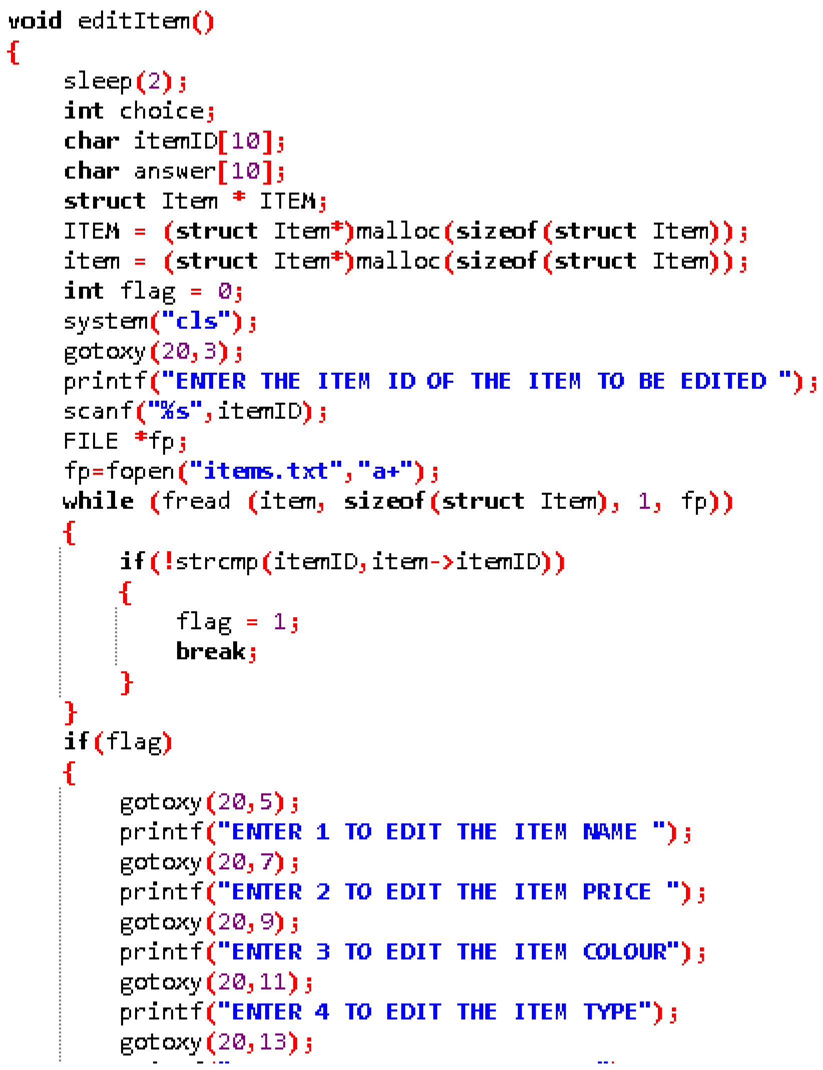
```
void addNewItem()
{
    sleep(2);
    system("cls");
    char temp[10];
    item = (struct Item*)malloc(sizeof(struct Item));
    FILE *fp;
    char itemID[10];
    char itemName[10];
    int itemPrice;
    char colour[10];
    char itemType[10];
    char itemBrand[10];
    fp=fopen("items.txt", "a+");
    gotoxy(20,3);
    printf("ENTER ITEM ID ");
    scanf("%s",temp);
    while (fread (item, sizeof(struct Item), 1, fp))
    {
        if(!strcmp(item->itemID,temp))
        {
            gotoxy(20,5);
            printf("ITEM IDs MATCH. PLEASE TRY AGAIN!");
            addNewItem();
        }
    }
    strcpy(item->itemID,temp);
    gotoxy(20,5);
    printf("ENTER ITEM NAME ");
    scanf("%s",&item->itemName);
    gotoxy(20,7);
    printf("ENTER ITEM PRICE ");
    scanf("%s",&item->itemPrice);
    gotoxy(20,9);
    printf("ENTER ITEM COLOUR ");
```

•

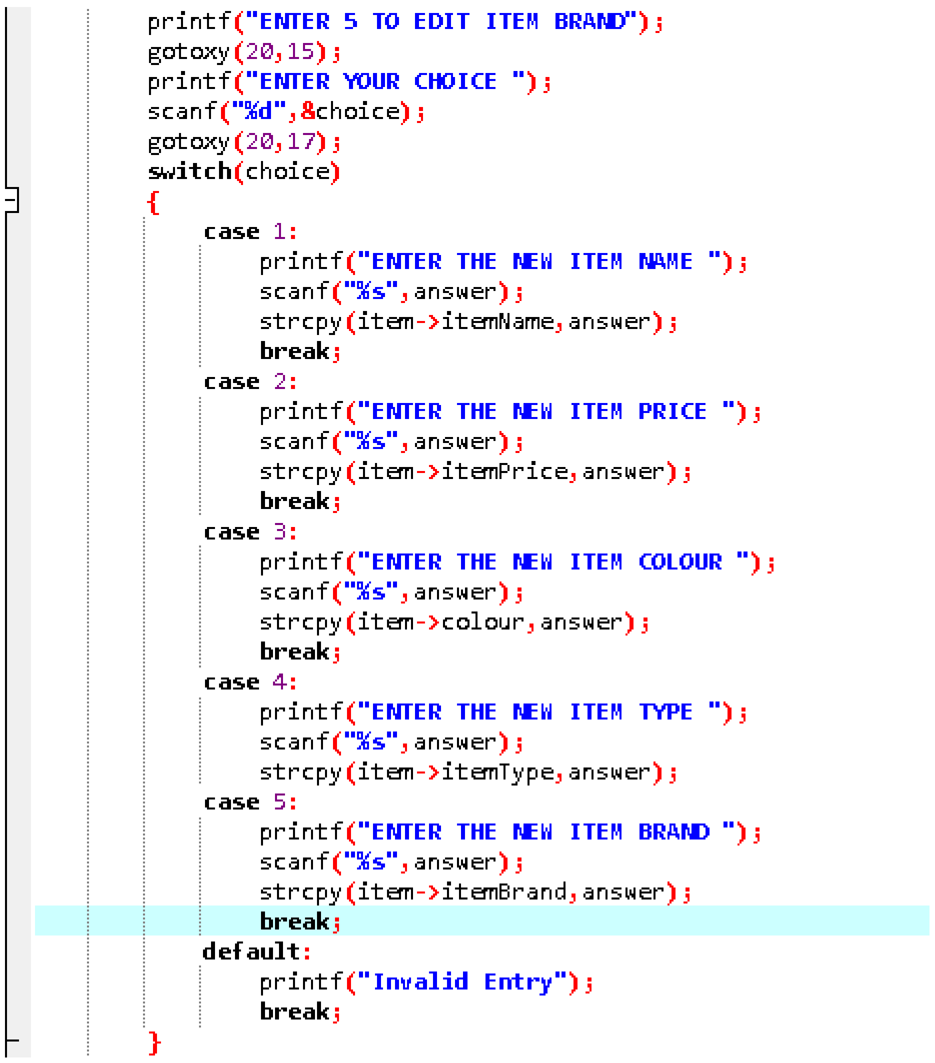


```
    printf("ENTER ITEM COLOUR ");
    scanf(" %s",&item->colour);
    gotoxy(20,11);
    printf("ENTER ITEM TYPE ");
    scanf("%s",&item->itemType);
    gotoxy(20,13);
    printf("ENTER ITEM BRAND ");
    scanf("%s",&item->itemBrand);
    fwrite (item, sizeof(struct Item), 1, fp);
    fclose(fp);
    gotoxy(20,15);
    printf("ITEM ADDED SUCCESSFULLY! PRESS ENTER TO CONTINUE");
    getch();
    free(item);
    return;
}
```

editItem()

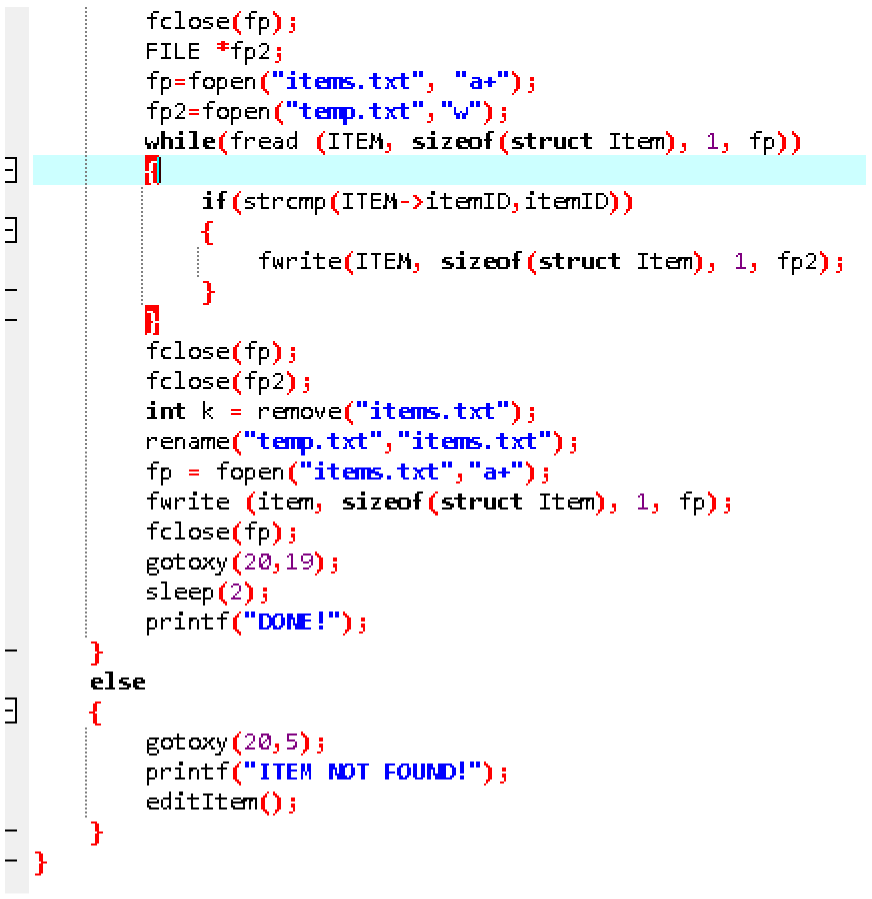

```c
void editItem()
{
    sleep(2);
    int choice;
    char itemID[10];
    char answer[10];
    struct Item * ITEM;
    ITEM = (struct Item*)malloc(sizeof(struct Item));
    item = (struct Item*)malloc(sizeof(struct Item));
    int flag = 0;
    system("cls");
    gotoxy(20,3);
    printf("ENTER THE ITEM ID OF THE ITEM TO BE EDITED ");
    scanf("%s",itemID);
    FILE *fp;
    fp=fopen("items.txt","a+");
    while (fread (item, sizeof(struct Item), 1, fp))
    {
        if(!strcmp(itemID,item->itemID))
        {
            flag = 1;
            break;
        }
    }
    if(flag)
    {
        gotoxy(20,5);
        printf("ENTER 1 TO EDIT THE ITEM NAME ");
        gotoxy(20,7);
        printf("ENTER 2 TO EDIT THE ITEM PRICE ");
        gotoxy(20,9);
        printf("ENTER 3 TO EDIT THE ITEM COLOUR");
        gotoxy(20,11);
        printf("ENTER 4 TO EDIT THE ITEM TYPE");
        gotoxy(20,13);
```

•



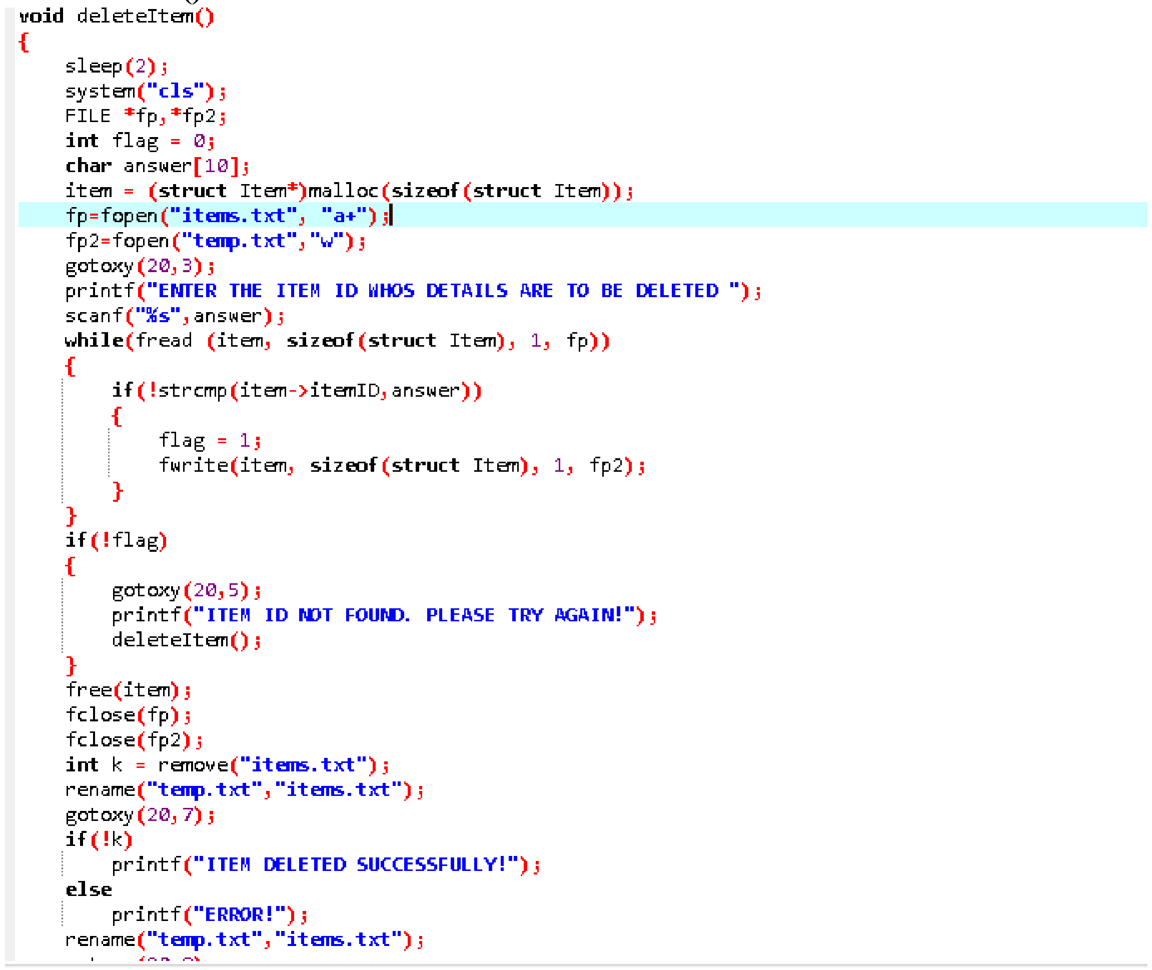
```
printf("ENTER 5 TO EDIT ITEM BRAND");
gotoxy(20,15);
printf("ENTER YOUR CHOICE ");
scanf("%d",&choice);
gotoxy(20,17);
switch(choice)
{
    case 1:
        printf("ENTER THE NEW ITEM NAME ");
        scanf("%s",answer);
        strcpy(item->itemName,answer);
        break;
    case 2:
        printf("ENTER THE NEW ITEM PRICE ");
        scanf("%s",answer);
        strcpy(item->itemPrice,answer);
        break;
    case 3:
        printf("ENTER THE NEW ITEM COLOUR ");
        scanf("%s",answer);
        strcpy(item->colour,answer);
        break;
    case 4:
        printf("ENTER THE NEW ITEM TYPE ");
        scanf("%s",answer);
        strcpy(item->itemType,answer);
    case 5:
        printf("ENTER THE NEW ITEM BRAND ");
        scanf("%s",answer);
        strcpy(item->itemBrand,answer);
        break;
    default:
        printf("Invalid Entry");
        break;
}
```

•



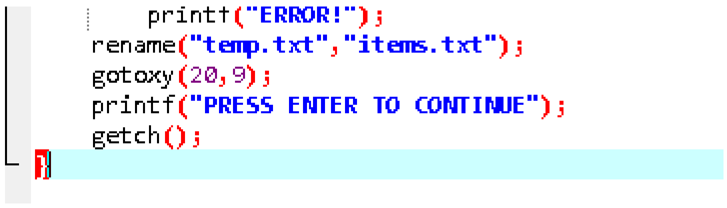
```
        fclose(fp);
        FILE *fp2;
        fp=fopen("items.txt", "a+");
        fp2=fopen("temp.txt","w");
        while(fread (ITEM, sizeof(struct Item), 1, fp))
        {
            if(strcmp(ITEM->itemID,itemID))
            {
                fwrite(ITEM, sizeof(struct Item), 1, fp2);
            }
        }
        fclose(fp);
        fclose(fp2);
        int k = remove("items.txt");
        rename("temp.txt","items.txt");
        fp = fopen("items.txt","a+");
        fwrite (item, sizeof(struct Item), 1, fp);
        fclose(fp);
        gotoxy(20,19);
        sleep(2);
        printf("DONE!");
    }
    else
    {
        gotoxy(20,5);
        printf("ITEM NOT FOUND!");
        editItem();
    }
}
```

deleteItem()



```
void deleteItem()
{
    sleep(2);
    system("cls");
    FILE *fp,*fp2;
    int flag = 0;
    char answer[10];
    item = (struct Item*)malloc(sizeof(struct Item));
    fp=fopen("items.txt", "a+");
    fp2=fopen("temp.txt","w");
    gotoxy(20,3);
    printf("ENTER THE ITEM ID WHOS DETAILS ARE TO BE DELETED ");
    scanf("%s",answer);
    while(fread (item, sizeof(struct Item), 1, fp))
    {
        if(!strcmp(item->itemID,answer))
        {
            flag = 1;
            fwrite(item, sizeof(struct Item), 1, fp2);
        }
    }
    if(!flag)
    {
        gotoxy(20,5);
        printf("ITEM ID NOT FOUND. PLEASE TRY AGAIN!");
        deleteItem();
    }
    free(item);
    fclose(fp);
    fclose(fp2);
    int k = remove("items.txt");
    rename("temp.txt","items.txt");
    gotoxy(20,7);
    if(!k)
        printf("ITEM DELETED SUCCESSFULLY!");
    else
        printf("ERROR!");
    rename("temp.txt","items.txt");
```

………

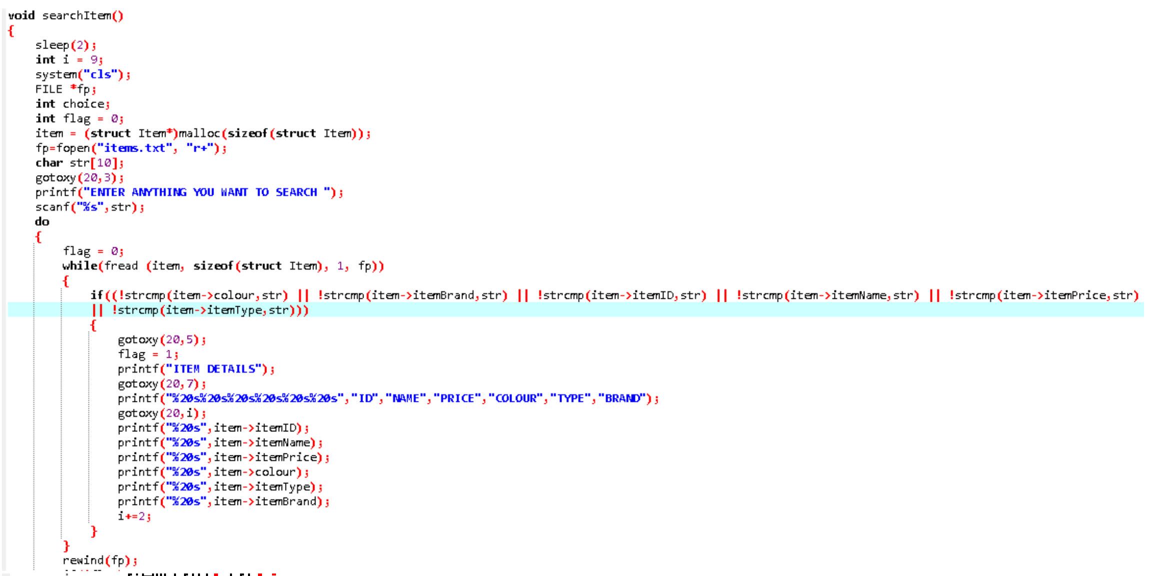


```
        printf("ERROR!");
    rename("temp.txt","items.txt");
    gotoxy(20,9);
    printf("PRESS ENTER TO CONTINUE");
    getch();
}
```

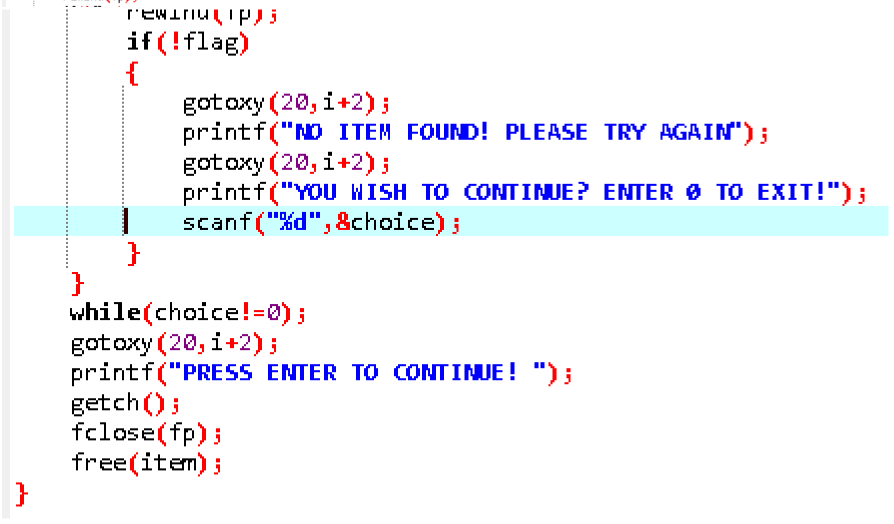
printItems()

```
void printItems()
{
    sleep(2);
    int i = 7;
    system("cls");
    FILE *fp;
    item = (struct Item*)malloc(sizeof(struct Item));
    fp=fopen("items.txt", "a+");
    gotoxy(20,3);
    printf("ITEM DETAILS");
    gotoxy(20,5);
    printf("%20s%20s%20s%20s%20s%20s","ID","NAME","PRICE","COLOUR","TYPE","BRAND");
    while(fread (item, sizeof(struct Item), 1, fp))
    {
        gotoxy(20,i);
        printf("%20s",item->itemID);
        printf("%20s",item->itemName);
        printf("%20s",item->itemPrice);
        printf("%20s",item->colour);
        printf("%20s",item->itemType);
        printf("%20s",item->itemBrand);
        i+=2;
    }
    gotoxy(20,i+2);
    printf("PRESS ENTER TO CONTINUE! ");
    getch();
    free(item);
    fclose(fp);
}
```

searchItem()

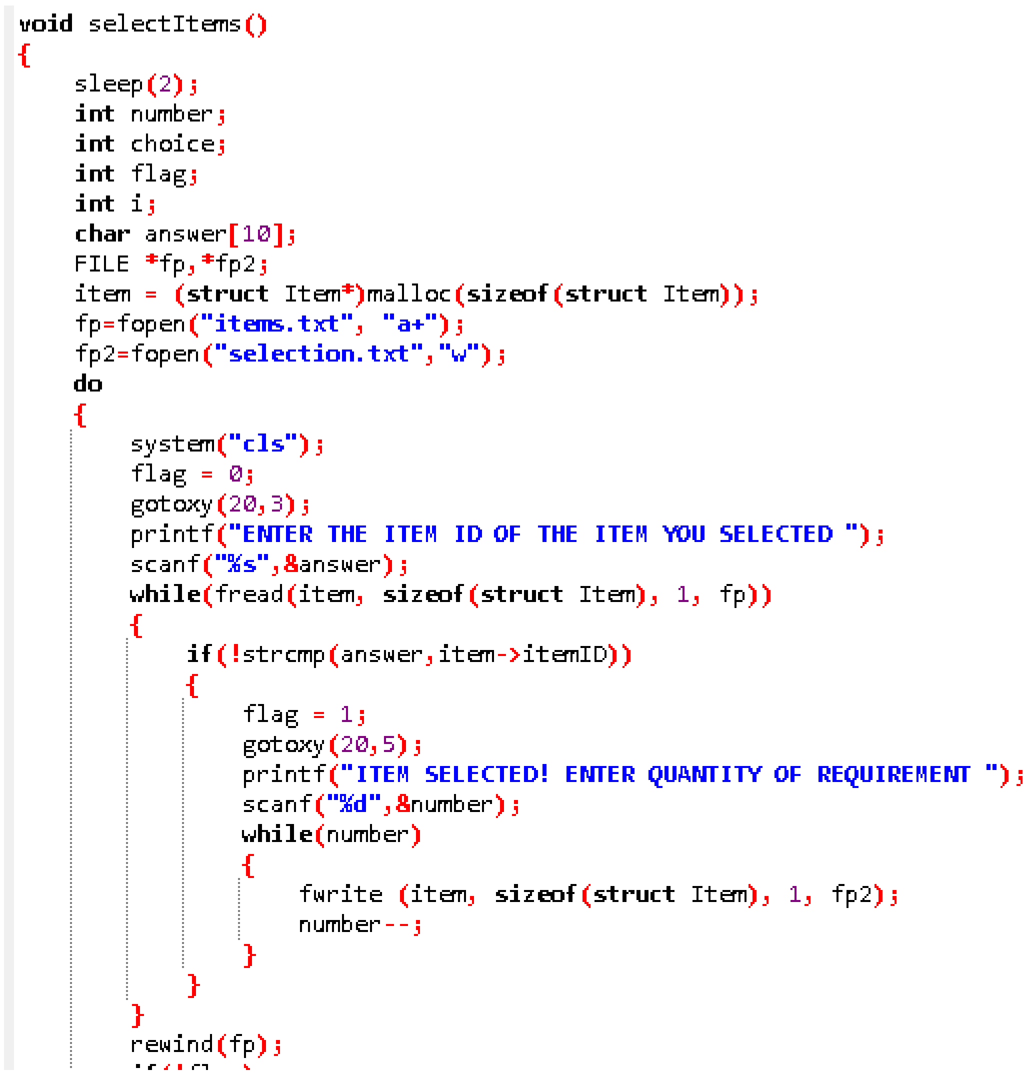


```c
void searchItem()
{
    sleep(2);
    int i = 9;
    system("cls");
    FILE *fp;
    int choice;
    int flag = 0;
    item = (struct Item*)malloc(sizeof(struct Item));
    fp=fopen("items.txt", "r+");
    char str[10];
    gotoxy(20,3);
    printf("ENTER ANYTHING YOU WANT TO SEARCH ");
    scanf("%s",str);
    do
    {
        flag = 0;
        while(fread (item, sizeof(struct Item), 1, fp))
        {
            if((!strcmp(item->colour,str) || !strcmp(item->itemBrand,str) || !strcmp(item->itemID,str) || !strcmp(item->itemName,str) || !strcmp(item->itemPrice,str)
            || !strcmp(item->itemType,str)))
            {
                gotoxy(20,5);
                flag = 1;
                printf("ITEM DETAILS");
                gotoxy(20,7);
                printf("%20s%20s%20s%20s%20s%20s","ID","NAME","PRICE","COLOUR","TYPE","BRAND");
                gotoxy(20,i);
                printf("%20s",item->itemID);
                printf("%20s",item->itemName);
                printf("%20s",item->itemPrice);
                printf("%20s",item->colour);
                printf("%20s",item->itemType);
                printf("%20s",item->itemBrand);
                i+=2;
            }
        }
        rewind(fp);
```

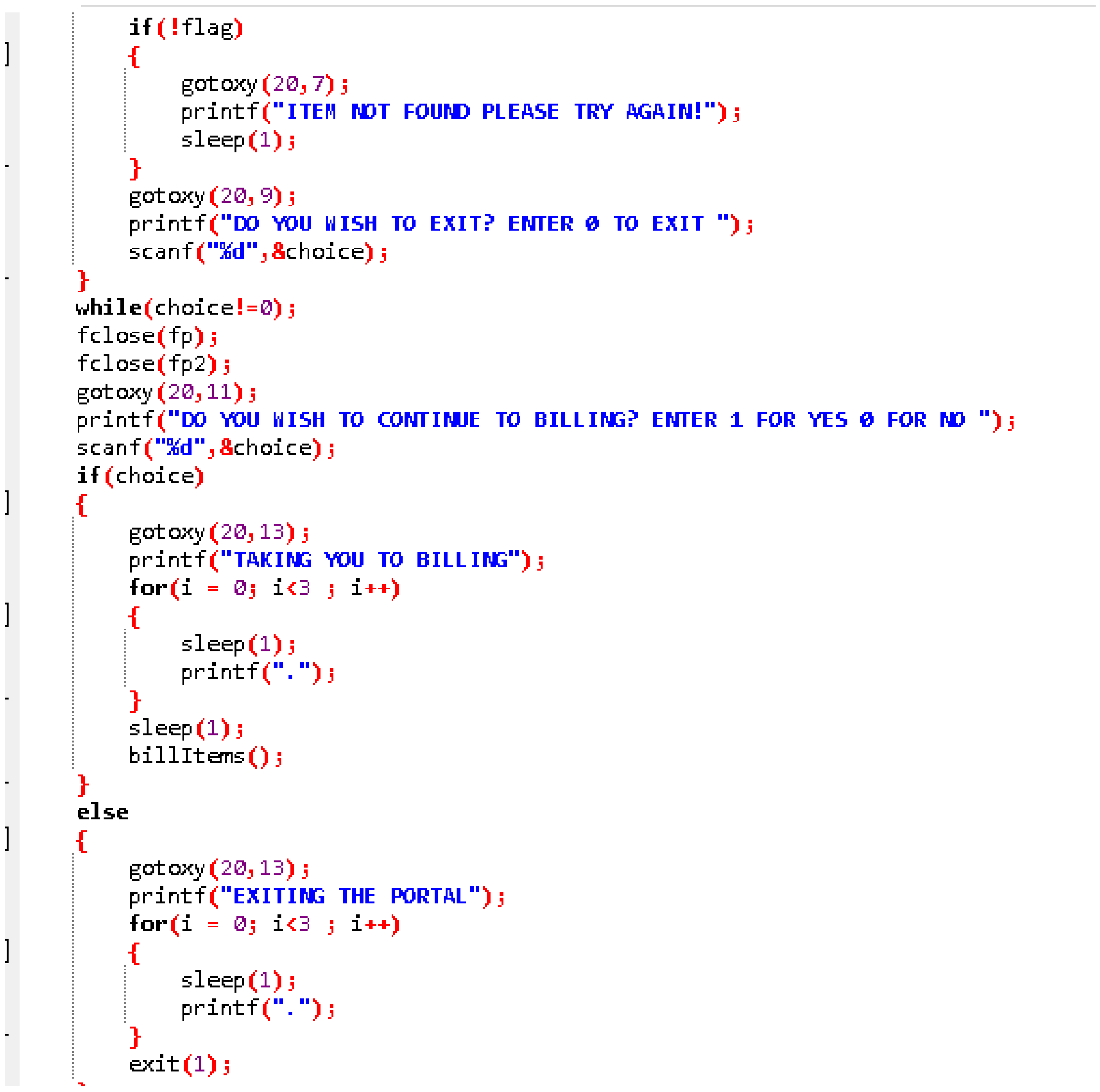


```c
        rewind(fp);
        if(!flag)
        {
            gotoxy(20,i+2);
            printf("NO ITEM FOUND! PLEASE TRY AGAIN");
            gotoxy(20,i+2);
            printf("YOU WISH TO CONTINUE? ENTER 0 TO EXIT!");
            scanf("%d",&choice);
        }
    }
    while(choice!=0);
    gotoxy(20,i+2);
    printf("PRESS ENTER TO CONTINUE! ");
    getch();
    fclose(fp);
    free(item);
}
```

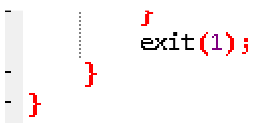
selectItems()



```
void selectItems()
{
    sleep(2);
    int number;
    int choice;
    int flag;
    int i;
    char answer[10];
    FILE *fp,*fp2;
    item = (struct Item*)malloc(sizeof(struct Item));
    fp=fopen("items.txt", "a+");
    fp2=fopen("selection.txt","w");
    do
    {
        system("cls");
        flag = 0;
        gotoxy(20,3);
        printf("ENTER THE ITEM ID OF THE ITEM YOU SELECTED ");
        scanf("%s",&answer);
        while(fread(item, sizeof(struct Item), 1, fp))
        {
            if(!strcmp(answer,item->itemID))
            {
                flag = 1;
                gotoxy(20,5);
                printf("ITEM SELECTED! ENTER QUANTITY OF REQUIREMENT ");
                scanf("%d",&number);
                while(number)
                {
                    fwrite (item, sizeof(struct Item), 1, fp2);
                    number--;
                }
            }
        }
        rewind(fp);
```

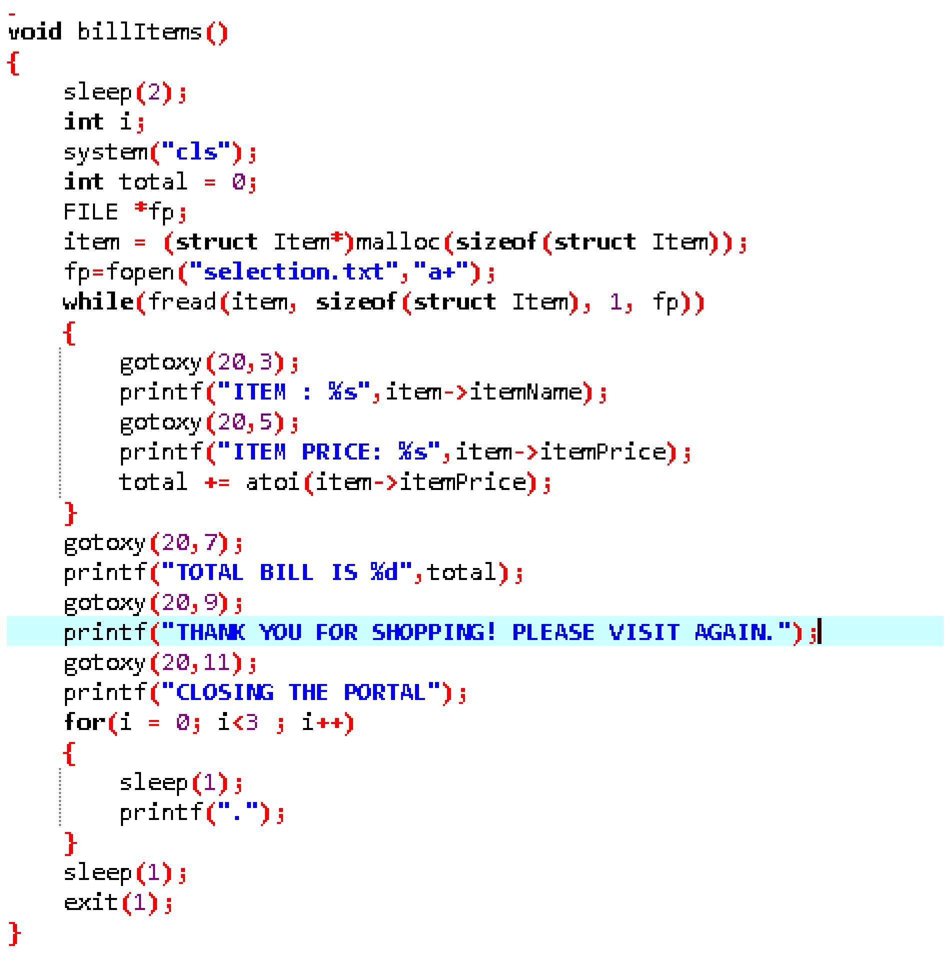


```
        if(!flag)
        {
            gotoxy(20,7);
            printf("ITEM NOT FOUND PLEASE TRY AGAIN!");
            sleep(1);
        }
        gotoxy(20,9);
        printf("DO YOU WISH TO EXIT? ENTER 0 TO EXIT ");
        scanf("%d",&choice);
    }
    while(choice!=0);
    fclose(fp);
    fclose(fp2);
    gotoxy(20,11);
    printf("DO YOU WISH TO CONTINUE TO BILLING? ENTER 1 FOR YES 0 FOR NO ");
    scanf("%d",&choice);
    if(choice)
    {
        gotoxy(20,13);
        printf("TAKING YOU TO BILLING");
        for(i = 0; i<3 ; i++)
        {
            sleep(1);
            printf(".");
        }
        sleep(1);
        billItems();
    }
    else
    {
        gotoxy(20,13);
        printf("EXITING THE PORTAL");
        for(i = 0; i<3 ; i++)
        {
            sleep(1);
            printf(".");
        }
        exit(1);
```



```
        }
        exit(1);
    }
}
```

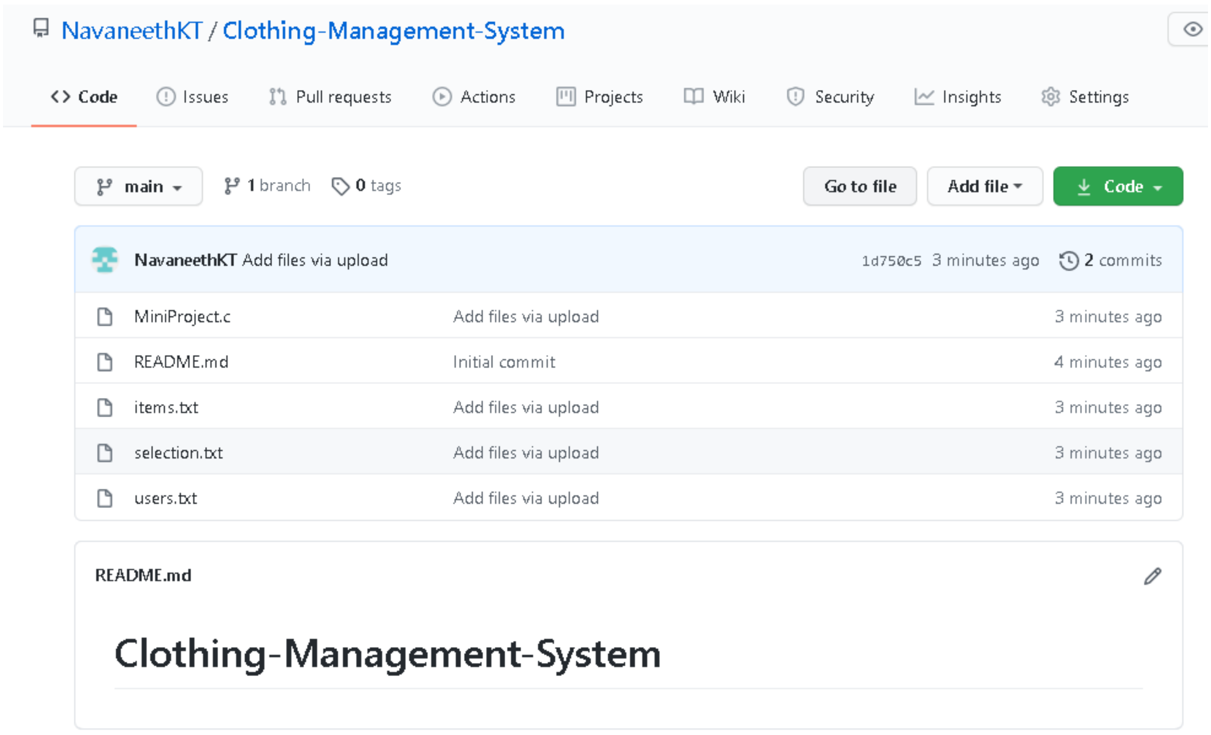
billItems()



```c
void billItems()
{
    sleep(2);
    int i;
    system("cls");
    int total = 0;
    FILE *fp;
    item = (struct Item*)malloc(sizeof(struct Item));
    fp=fopen("selection.txt","a+");
    while(fread(item, sizeof(struct Item), 1, fp))
    {
        gotoxy(20,3);
        printf("ITEM : %s",item->itemName);
        gotoxy(20,5);
        printf("ITEM PRICE: %s",item->itemPrice);
        total += atoi(item->itemPrice);
    }
    gotoxy(20,7);
    printf("TOTAL BILL IS %d",total);
    gotoxy(20,9);
    printf("THANK YOU FOR SHOPPING! PLEASE VISIT AGAIN.");
    gotoxy(20,11);
    printf("CLOSING THE PORTAL");
    for(i = 0; i<3 ; i++)
    {
        sleep(1);
        printf(".");
    }
    sleep(1);
    exit(1);
}
```

3.2.2. GITHUB/FOLDER STRUCTURE

All the text files and Source code is found in GitHub.



NavaneethKT Add files via upload 1d750c5 3 minutes ago 2 commits

MiniProject.c	Add files via upload	3 minutes ago
README.md	Initial commit	4 minutes ago
items.txt	Add files via upload	3 minutes ago
selection.txt	Add files via upload	3 minutes ago
users.txt	Add files via upload	3 minutes ago

README.md

Clothing-Management-System

3.3. TESTING

For the sake of simplicity, there is only one item in our inventory. Details of the items are

Item ID – S1

Item Name – CLOUDFOAM

Item Price – 3000

Item Colour – White

Item Type – Shoes

Item Brand - ADIDAS

3.3.1. USER USE CASES

User has 6 functionalities: Register, Login, Search item, Select Items, View Items and Bill Items.

REGISTER

Test Case Template		
Test Case ID: TC01		**User Case ID:** UC01
Test Case Title: Registering		
Test Case Description: User tries to register as a user with an already existing username.		
Test Steps:	**Expected Result:**	**Actual Result:**
1. System displays prompt for user to enter username 2. User enters a username which already exists.	System should display "Invalid credentials".	The System displays "USERNAME ALREADY EXISTS PLEASE TRY AGAIN".

Test Case Template		
Test Case ID: TC02		**User Case ID:** UC01
Test Case Title: Registering		
Test Case Description: User tries to register with a unique username.		
Test Steps:	**Expected Result:**	**Actual Result:**
1. System displays prompt for user to enter password. 2. User enters a unique username.	System should display “Login successful”	System displays “Taking you to the login screen…”

LOGIN

Test Case Template		
Test Case ID: TC03		**User Case ID:** UC02
Test Case Title: Logging in		
Test Case Description: User enters an invalid password		
Test Steps:	**Expected Result:**	**Actual Result:**
1. System displays prompt for user to enter username and password. 2. User enters the wrong the password.	System should display an error message and should repeat the login process.	The System displays “INVALID PASSWORD PLEASE TRY AGAIN” and prompts the user to re-enter username and password.

Test Case Template		
Test Case ID: TC04		**User Case ID:** UC02
Test Case Title: Logging in		
Test Case Description: User enters the correct password		
Test Steps:	**Expected Result:**	**Actual Result:**
1. System displays prompt for user to enter username and password 2. User enters a username and password which are valid.	System should display message "Login successful".	System displays "LOGIN SUCCESSFUL!" and displays date and time of login and takes the User- to, user page

SEARCH ITEM

Test Case Template		
Test Case ID: TC05		**User Case ID:**
Test Case Title: Searching an item		
Test Case Description: User attempts to search an existing item in the stores inventory.		UC03
Test Steps:	**Expected Result:**	**Actual Result:**
1. System displays list of User features 2. User selects search item. 3. System asks the user to enter a keyword 4.User enters keyword which isn't in the inventory.	System should display "Item not found".	System displays "NO ITEM FOUND PLEASE TRY AGAIN". And asks the user if they want to try again or exit the search page.

Test Case Template		
Test Case ID: TC06		**User Case ID:** UC03
Test Case Title: Searching an item		
Test Case Description User attempts to search an existing item in the stores inventory.		
Test Steps:	**Expected Result:**	**Actual Result:**
1. System displays list of User features 2. User selects search item. 3. System asks the user to enter a keyword 4.User enters keyword which is in the inventory.	System should display the details of the item.	System displays the entire details of the item

VIEW ITEMS

Test Case Template		
Test Case ID: TC07		**User Case ID:** UC04
Test Case Title: View Items		
Test Case Description: User attempts to view all items present in the inventory		
Test Steps:	**Expected Result:**	**Actual Result:**
1. System displays list of User features 2. User selects view Items.	System should print all items details of all the items present	System prints all items details of all the items present

SELECT ITEMS

Test Case Template		
Test Case ID: TC08		**User Case ID:**
Test Case Title: Select Items		
Test Case Description: User attempts to select few items.		UC05
Test Steps:	**Expected Result:**	**Actual Result:**
1. System displays list of User features 2. User selects select items 3. System prompts the user to enter the item ID of the item which he selected 4. User enters wrong Item ID	System should display item not found	System displays "ITEM NOT FOUND PLEASE TRY AGAIN"

Test Case Template		
Test Case ID: TC09		**User Case ID:** UC05
Test Case Title: Select Items		
Test Case Description: User attempts to select few items and head towards billing		
Test Steps:	**Expected Result:**	**Actual Result:**
1. System displays list of User features 2. User selects select items 3. System prompts the user to enter the item ID of the item which he selected 4. User enters wrong Item ID 5. System prompts the user to enter quantity of requirement 6. User enters quantity of requirement 7. System asks if the User wants to head to check out 8. User agrees to checkout	Selected items along with quantity are selected	Selected items along with quantity are selected proceed to billing

Test Case Template		
Test Case ID: TC10		**User Case ID:** UC05
Test Case Title: Select Items		
Test Case Description: User attempts to select few items and doesn't head towards billing		
Test Steps:	**Expected Result:**	**Actual Result:**
1. System displays list of User features 2. User selects select items 3. System prompts the user to enter the item ID of the item which he selected 4. User enters wrong Item ID 5. System prompts the user to enter quantity of requirement 6. User enters quantity of requirement 7. System asks if the User wants to head to check out 8. User disagrees to checkout	Selected items along with quantity are selected	Selected items along with quantity are selected but the portal is terminated

BILLING ITEMS

Test Case Template		
Test Case ID: TC11		**User Case ID:** UC06
Test Case Title: Bill Items		
Test Case Description: User proceeds to checkout and waits for the bill		
Test Steps:	**Expected Result:**	**Actual Result:**
1. User agrees to checkout from Select Items page. 2. System takes the user to billing page	Total bill is calculated.	Total bill along with other item details are printed

3.3.3. ADMIN TEST CASES

Admin has 6 functionalities. Login, Adding New Items, Editing Items, Deleting Items, Search Items and View Items

LOGIN

Test Case Template		
Test Case ID: TC12		**User Case ID:** UC07
Test Case Title: Log in		
Test Case Description: Admin attempts to login with incorrect password		
Test Steps:	**Expected Result:**	**Actual Result:**
1. System displays prompt for Admin to login. 2. Admin enters invalid password.	System should display “Wrong password”	System displays “INVALID PASSWORD PLEASE TRY AGAIN!” and restarts the admin login page

Test Case Template		
Test Case ID: TC13		**User Case ID:** UC07
Test Case Title: Log in		
Test Case Description: Admin attempts to login with correct login credentials		
Test Steps:	**Expected Result:**	**Actual Result:**
1. System displays prompt for Admin to login. 2. Admin enters correct login credentials	System should display admin features.	System displays “LOGIN SUCCESSFUL” and lists all the admin features

ADDING ITEM

Test Case Template		
Test Case ID: TC14		**User Case ID:** UC08
Test Case Title: Adding a new item		
Test Case Description: Admin attempts to add a new item.		
Test Steps:	**Expected Result:**	**Actual Result:**
1. System displays Admin features 2. Admin selects adding item	System should prompt the admin details for the new item	System prompts the admin details for the new item

EDITING ITEM

Test Case Template		
Test Case ID: TC15		**User Case ID:** UC09
Test Case Title: Editing Item		
Test Case Description: Admin attempts to edit details of an item who item ID doesn't exist in the inventory		
Test Steps:	**Expected Result:**	**Actual Result:**
1. System displays Admin features 2. Admin selects editing item 3. System prompts the item ID of the item whose details are to be edited 4. Admin enters item ID which doesn't exist.	System should display invalid item ID	System displays "ITEM NOT FOUND" and restarts the edit page

Test Case Template		
Test Case ID: TC16		**User Case ID:** UC09
Test Case Title: Editing Item		
Test Case Description: Admin attempts to edit details of an item who item ID exists in the inventory		
Test Steps:	**Expected Result:**	**Actual Result:**
1. System displays Admin features 2. Admin selects editing item 3. System prompts the item ID of the item whose details are to be edited 4. Admin enters item ID which doesn't exist.	System should display "which detail of the item is to be edited".	System displays list of Item details and asks which one is to be edited

DELETE ITEM

Test Case Template		
Test Case ID: TC17		**User Case ID:** UC10
Test Case Title: Deleting Item		
Test Case Description: Admin attempts to delete an item which doesn’t exist in the inventory		
Test Steps:	**Expected Result:**	**Actual Result:**
1. System displays Admin features 2. Admin selects delete item 3. System prompts the item ID of the item which is to be deleted 4. Admin enters item ID which doesn’t exist.	System should display “item id doesn’t exist”	System displays “ITEM ID NOT FOUND PLEASE TRY AGAIN” and restarts the delete page

Test Case Template		
Test Case ID: TC18		**User Case ID:** UC10
Test Case Title: Deleting Item		
Test Case Description: Admin attempts to delete an item which doesn’t exist in the inventory		
Test Steps:	**Expected Result:**	**Actual Result:**
1. System displays Admin features 2. Admin selects delete item 3. System prompts the item ID of the item which is to be deleted 4. Admin enters item ID which exists	System should display ” item deleted successfully”	System displays “ITEM DELETED SUCCESSFULLY!”

SEARCH ITEM

Test Case Template		
Test Case ID: TC19		**User Case ID:**
Test Case Title: Searching an item		
Test Case Description: Admin attempts to search an existing item in the stores inventory.		UC11
Test Steps:	**Expected Result:**	**Actual Result:**
1. System displays list of Admin features 2. Admin selects search item. 3. System asks the Admin to enter a keyword 4. Admin enters keyword which isn't in the inventory.	System should display "Item not found".	System displays "NO ITEM FOUND PLEASE TRY AGAIN". And asks the admin if they want to try again or exit the search page.

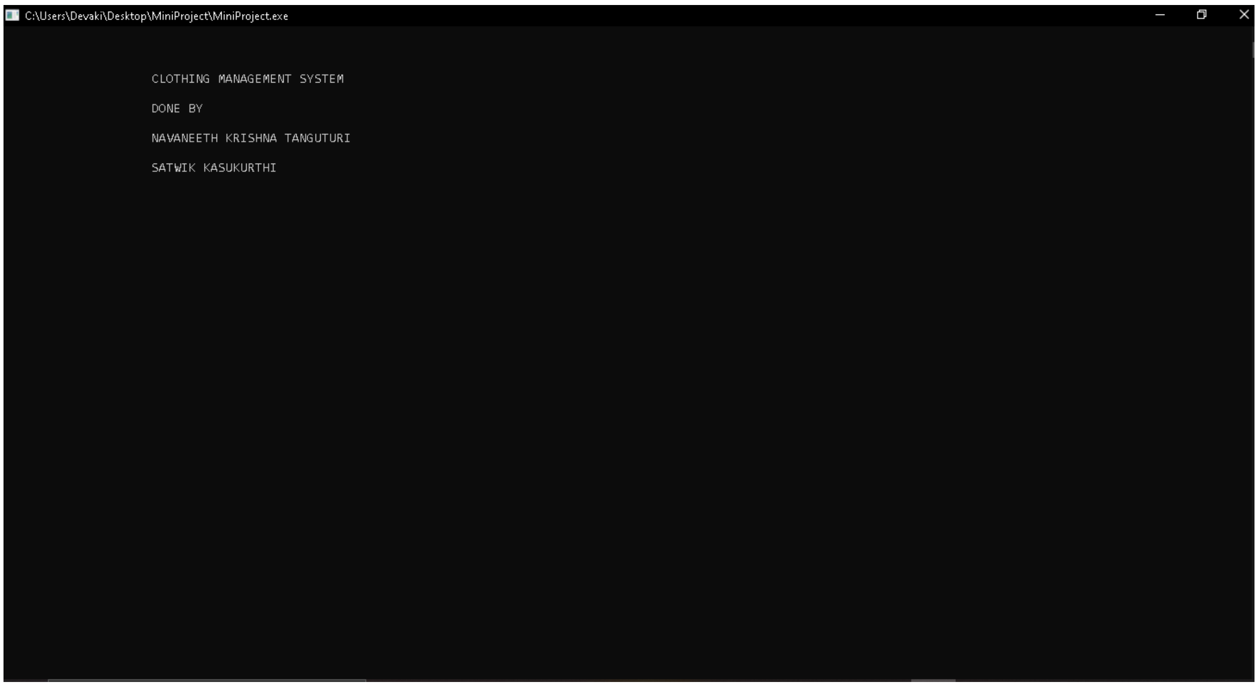
Test Case Template		
Test Case ID: TC20		**User Case ID:** UC11
Test Case Title: Searching an item		
Test Case Description Admin attempts to search an existing item in the stores inventory.		
Test Steps:	**Expected Result:**	**Actual Result:**
1. System displays list of Admin features 2. Admin selects search item. 3. System asks the Admin to enter a keyword 4. Admin enters keyword which isn't in the inventory.	System should display the details of the item.	System displays the entire details of the item

VIEW ITEMS

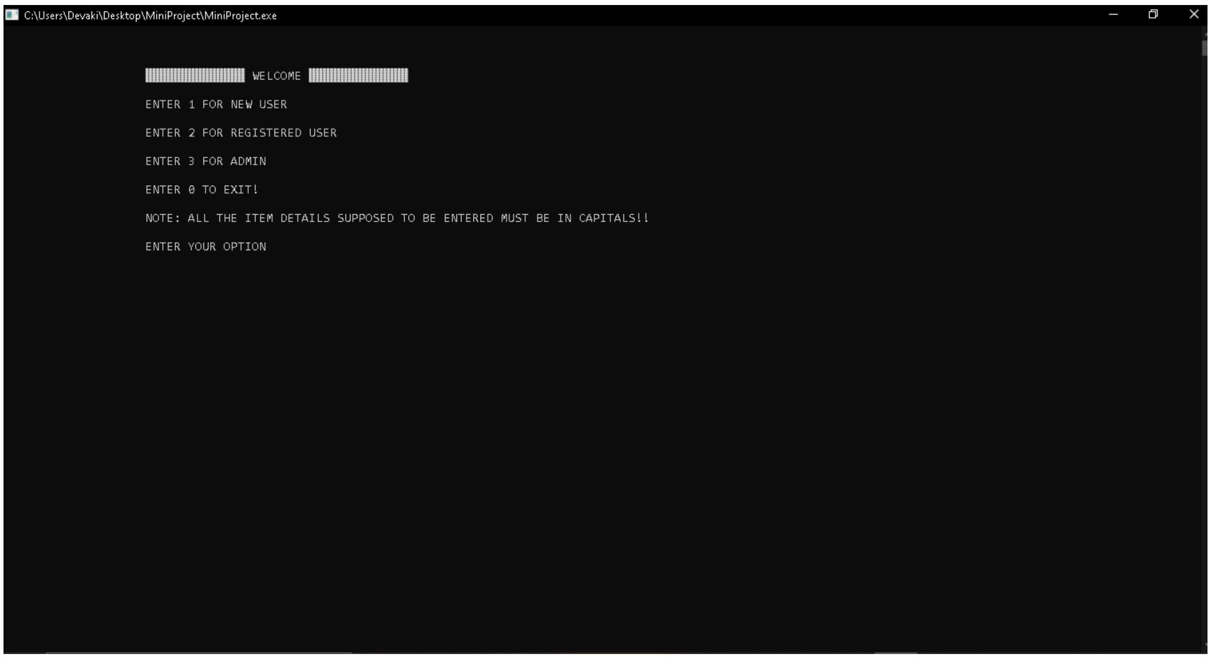
Test Case Template		
Test Case ID: TC21		**User Case ID:** UC12
Test Case Title: View Items		
Test Case Description: Admin attempts to view all items present in the inventory		
Test Steps:	**Expected Result:**	**Actual Result:**
1. System displays list of Admin features 2. Admin selects view Items.	System should print all items details of all the items present	System prints all items details of all the items present

4. RESULTS

WELCOME SCREEN



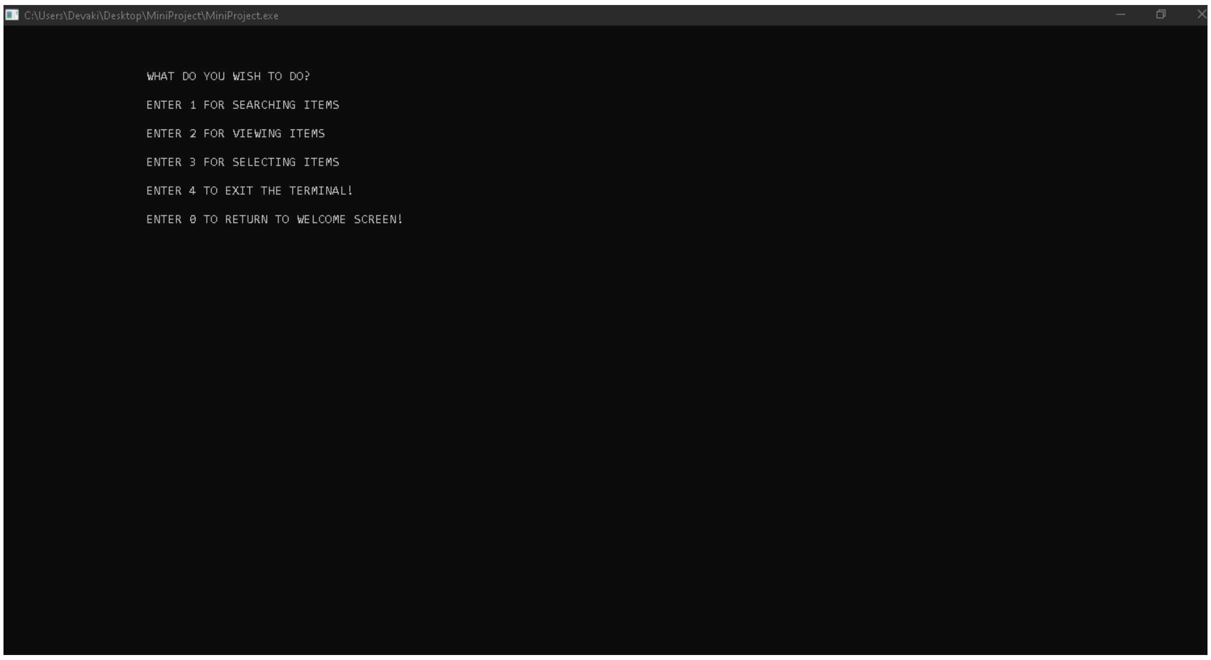
```
C:\Users\Devaki\Desktop\MiniProject\MiniProject.exe

                CLOTHING MANAGEMENT SYSTEM

                DONE BY

                NAVANEETH KRISHNA TANGUTURI

                SATWIK KASUKURTHI
```



```
C:\Users\Devaki\Desktop\MiniProject\MiniProject.exe

              ▒▒▒▒▒▒▒▒▒▒▒▒▒▒ WELCOME ▒▒▒▒▒▒▒▒▒▒▒▒▒▒
              ENTER 1 FOR NEW USER
              ENTER 2 FOR REGISTERED USER
              ENTER 3 FOR ADMIN
              ENTER 0 TO EXIT!
              NOTE: ALL THE ITEM DETAILS SUPPOSED TO BE ENTERED MUST BE IN CAPITALS!!
              ENTER YOUR OPTION
```

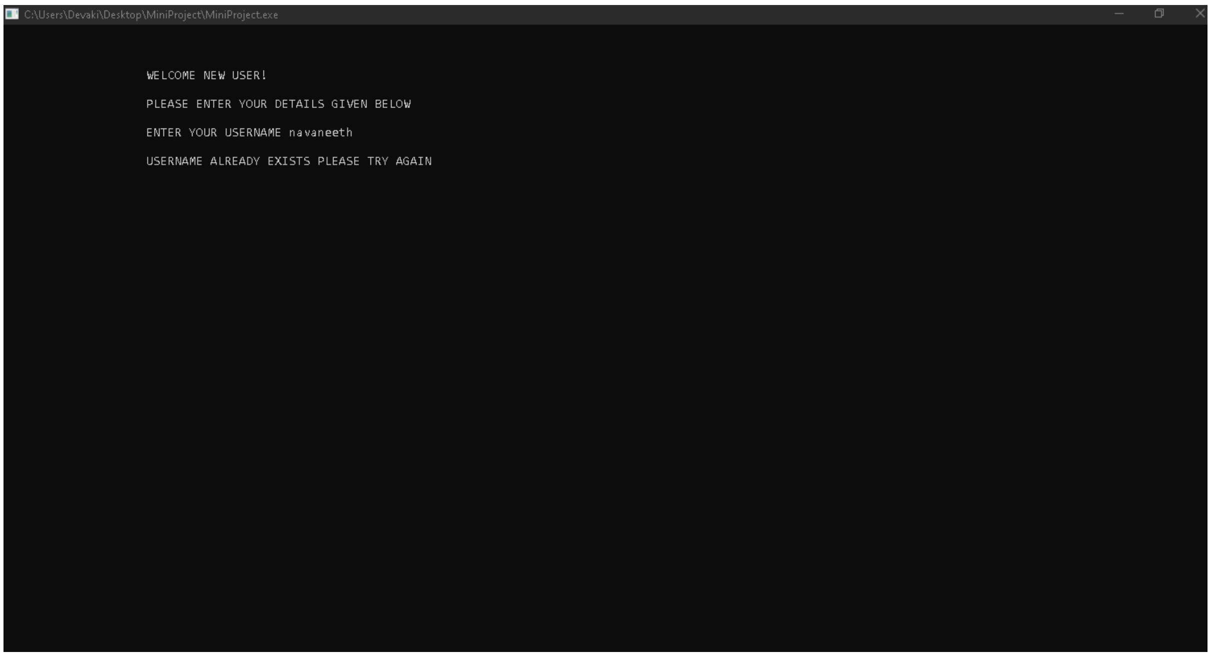
4.1 USER USE CASES

User Menu Screen



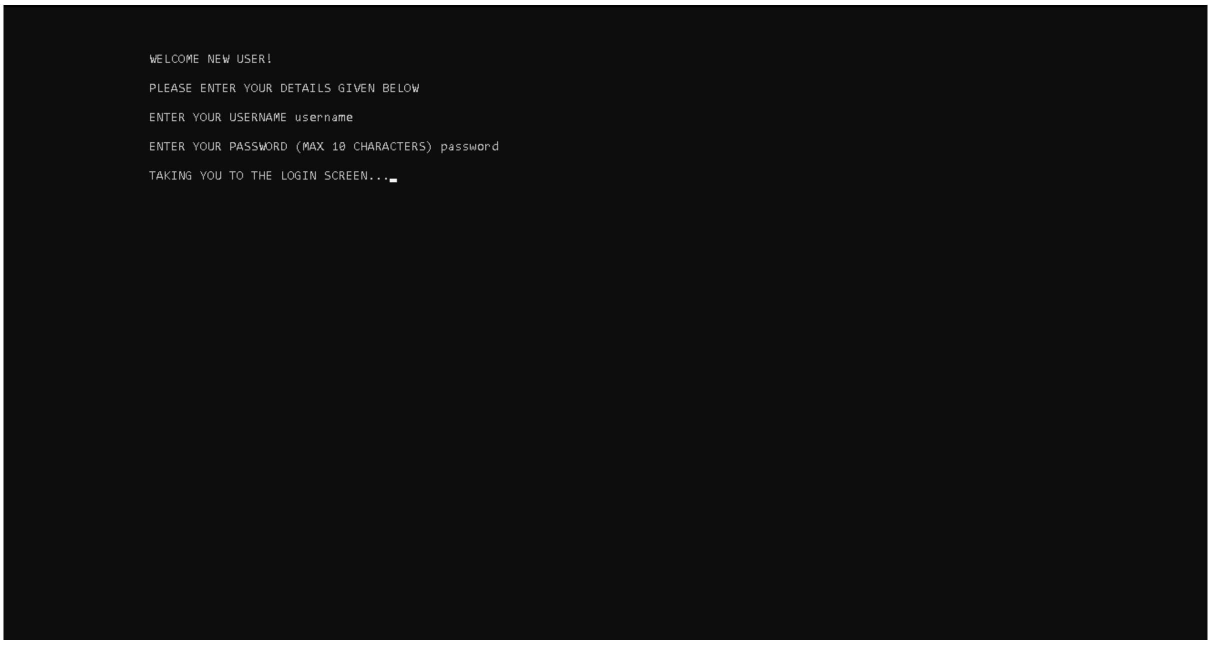
```
C:\Users\Devaki\Desktop\MiniProject\MiniProject.exe

              WHAT DO YOU WISH TO DO?
              ENTER 1 FOR SEARCHING ITEMS
              ENTER 2 FOR VIEWING ITEMS
              ENTER 3 FOR SELECTING ITEMS
              ENTER 4 TO EXIT THE TERMINAL!
              ENTER 0 TO RETURN TO WELCOME SCREEN!
```

Test Case 1



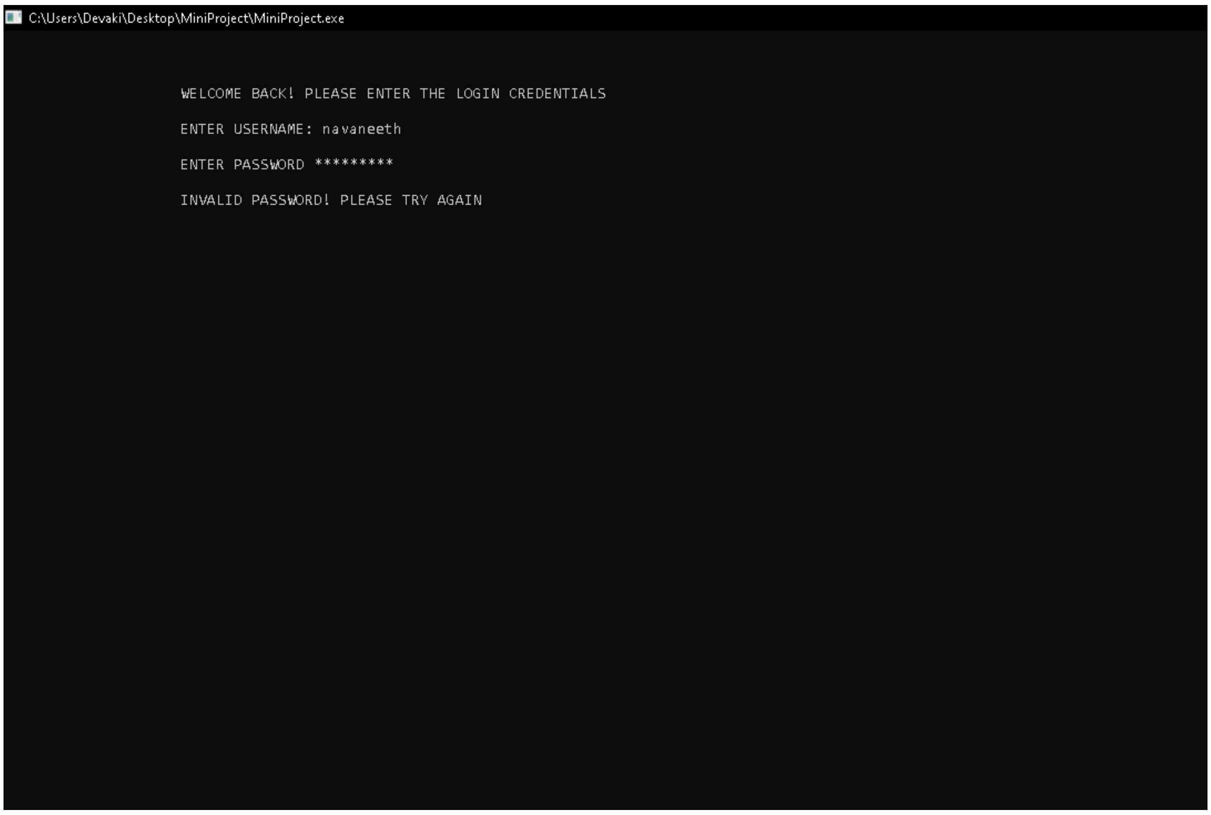
```
C:\Users\Devaki\Desktop\MiniProject\MiniProject.exe

WELCOME NEW USER!

PLEASE ENTER YOUR DETAILS GIVEN BELOW

ENTER YOUR USERNAME navaneeth

USERNAME ALREADY EXISTS PLEASE TRY AGAIN
```

Test Case 2



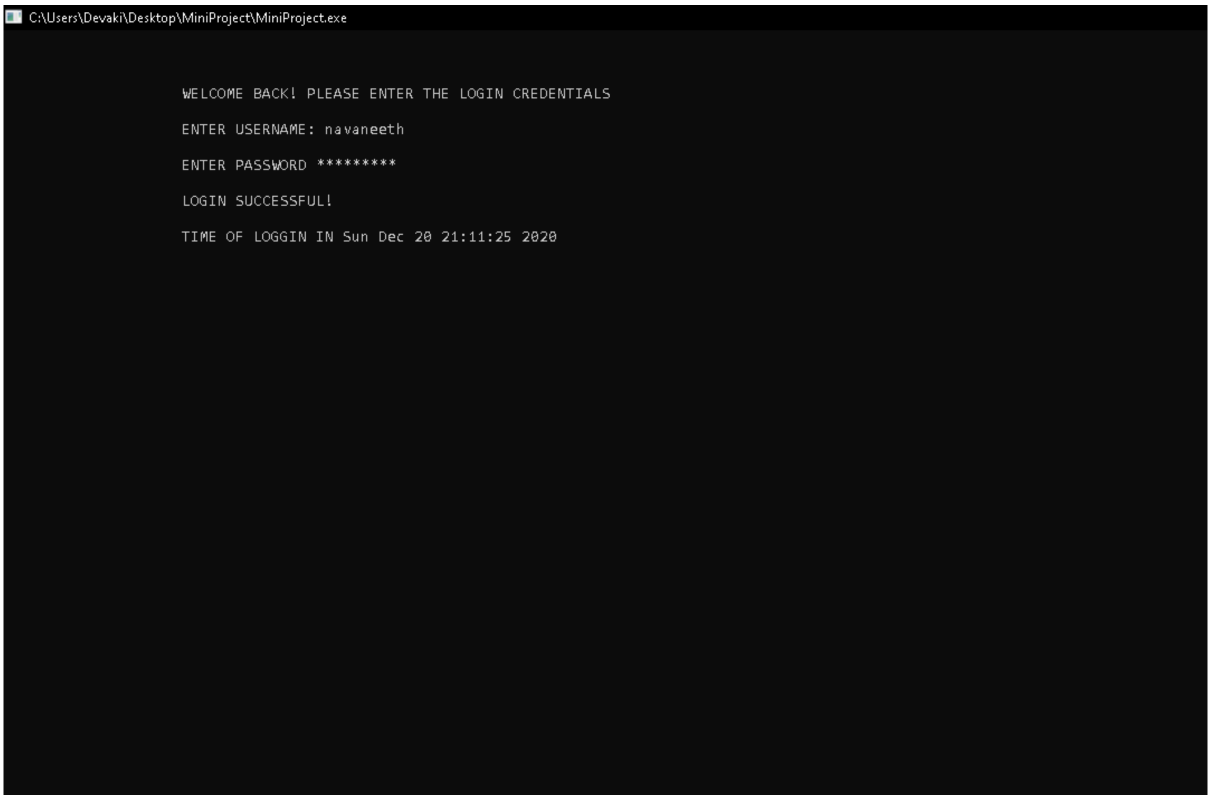
```
WELCOME NEW USER!

PLEASE ENTER YOUR DETAILS GIVEN BELOW

ENTER YOUR USERNAME username

ENTER YOUR PASSWORD (MAX 10 CHARACTERS) password

TAKING YOU TO THE LOGIN SCREEN...
```

Test Case 3



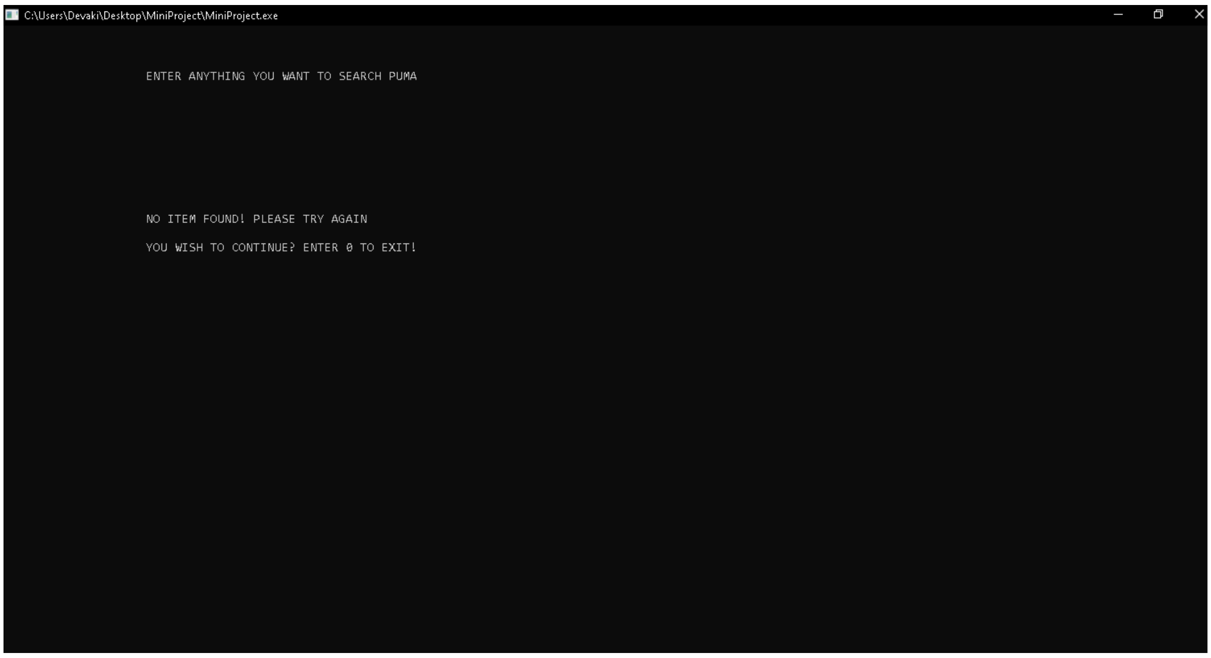
```
C:\Users\Devaki\Desktop\MiniProject\MiniProject.exe

                    WELCOME BACK! PLEASE ENTER THE LOGIN CREDENTIALS

                    ENTER USERNAME: navaneeth

                    ENTER PASSWORD *********

                    INVALID PASSWORD! PLEASE TRY AGAIN
```

Test Case 4



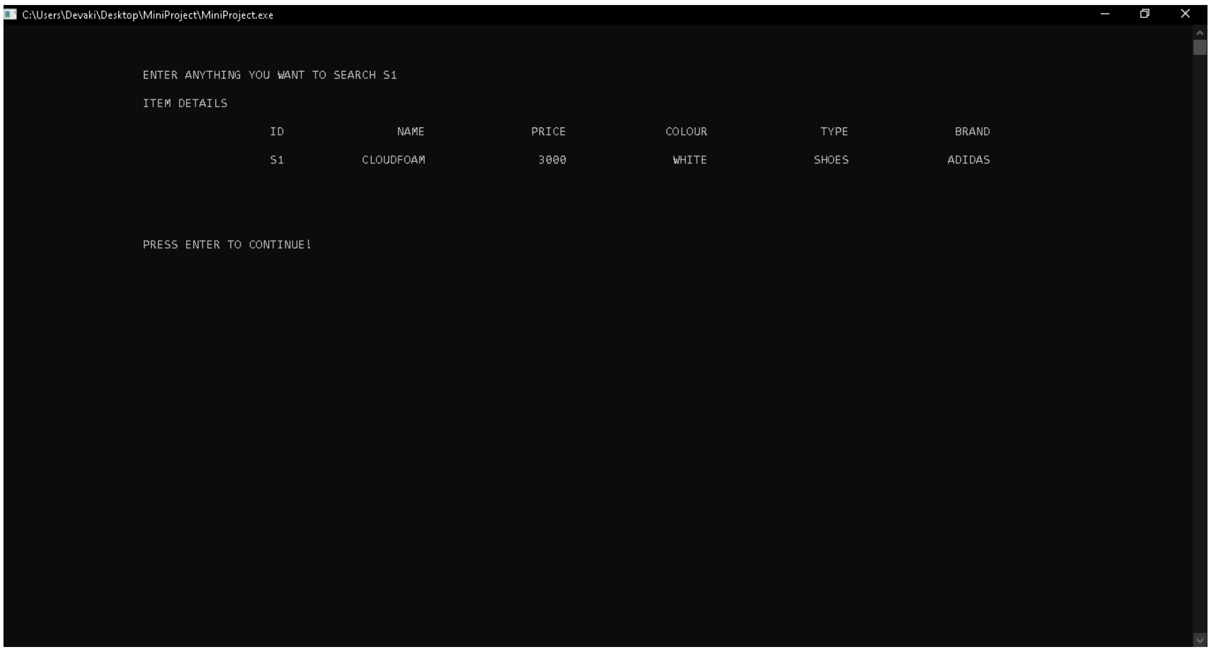
```
C:\Users\Devaki\Desktop\MiniProject\MiniProject.exe

                    WELCOME BACK! PLEASE ENTER THE LOGIN CREDENTIALS

                    ENTER USERNAME: navaneeth

                    ENTER PASSWORD *********

                    LOGIN SUCCESSFUL!

                    TIME OF LOGGIN IN Sun Dec 20 21:11:25 2020
```

Test Case 5



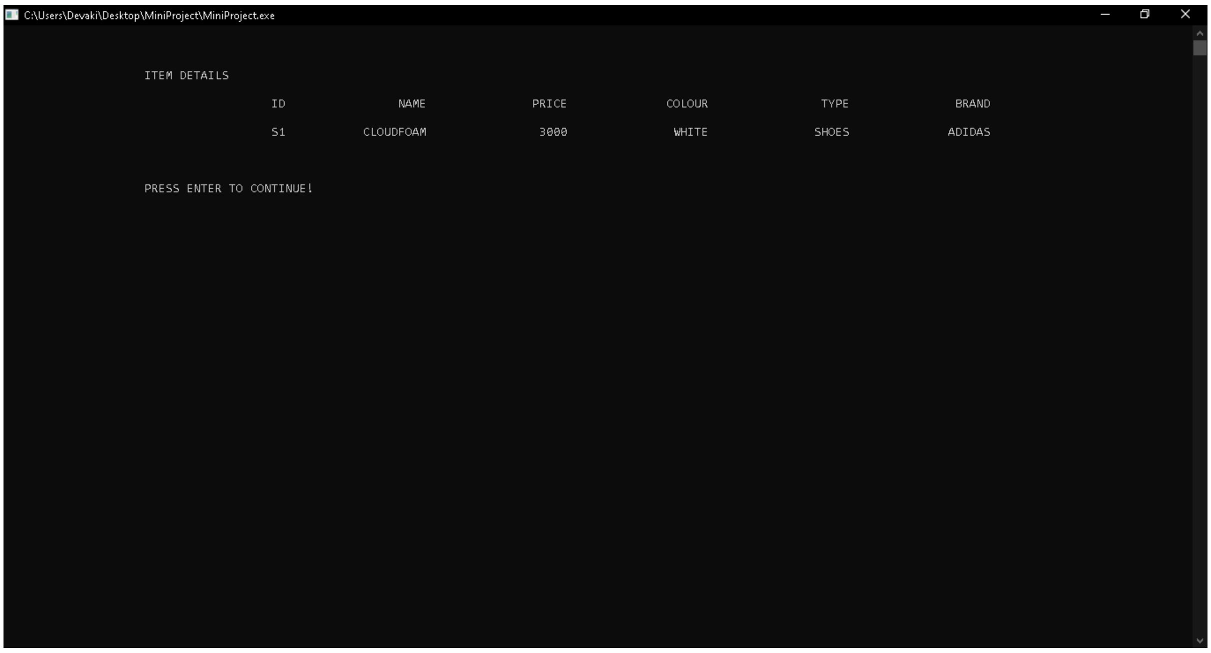
```
C:\Users\Devaki\Desktop\MiniProject\MiniProject.exe

                ENTER ANYTHING YOU WANT TO SEARCH PUMA




                NO ITEM FOUND! PLEASE TRY AGAIN

                YOU WISH TO CONTINUE? ENTER 0 TO EXIT!
```

Test Case 6



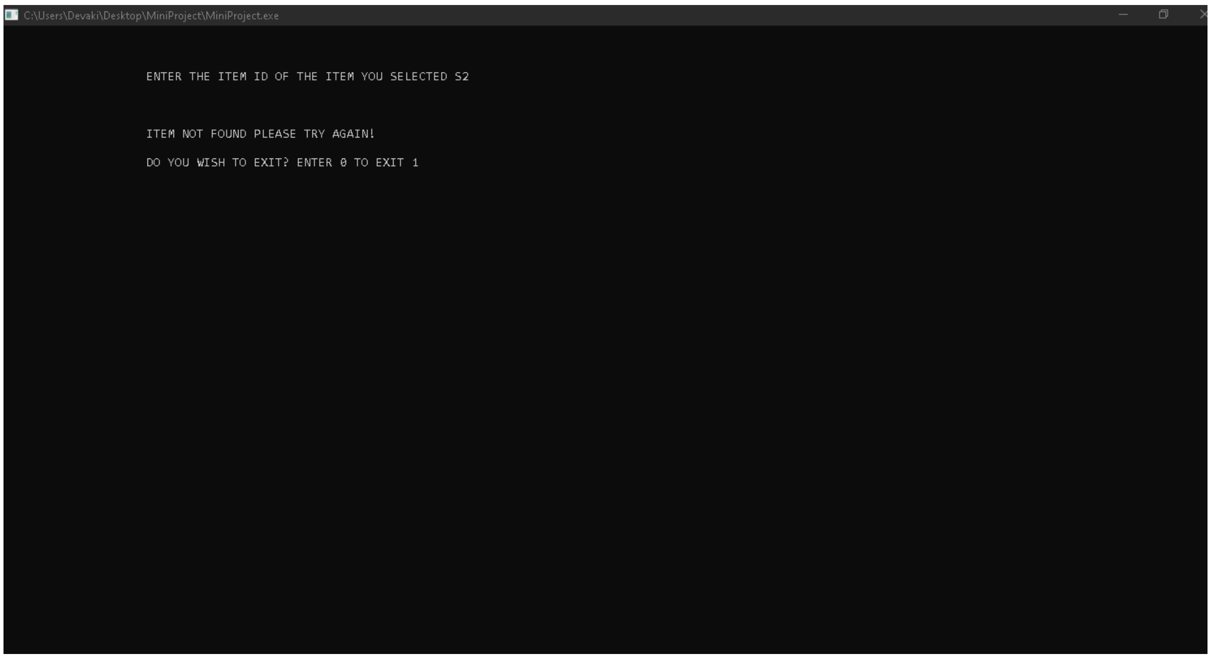
```
C:\Users\Devaki\Desktop\MiniProject\MiniProject.exe

                ENTER ANYTHING YOU WANT TO SEARCH S1

                ITEM DETAILS

                               ID              NAME            PRICE            COLOUR             TYPE             BRAND

                               S1          CLOUDFOAM            3000             WHITE            SHOES            ADIDAS



                PRESS ENTER TO CONTINUE!
```

Test Case 7



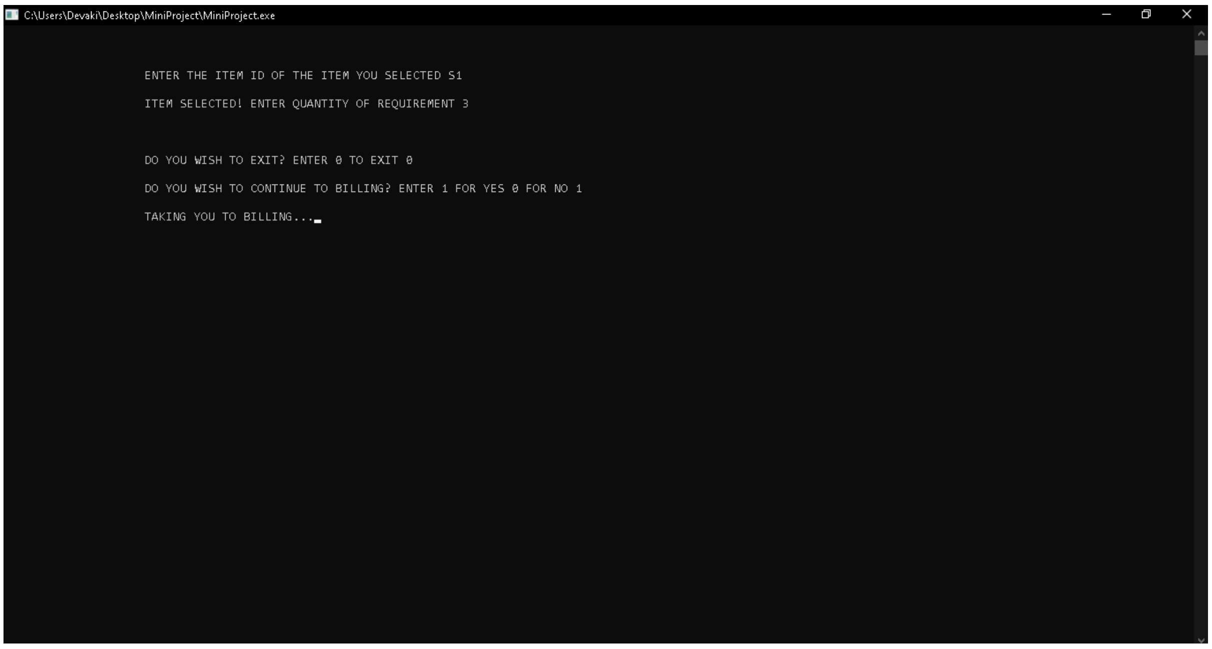
```
C:\Users\Devaki\Desktop\MiniProject\MiniProject.exe

ITEM DETAILS
                ID              NAME              PRICE              COLOUR              TYPE              BRAND
                S1          CLOUDFOAM              3000               WHITE             SHOES             ADIDAS


PRESS ENTER TO CONTINUE!
```

Test Case 8



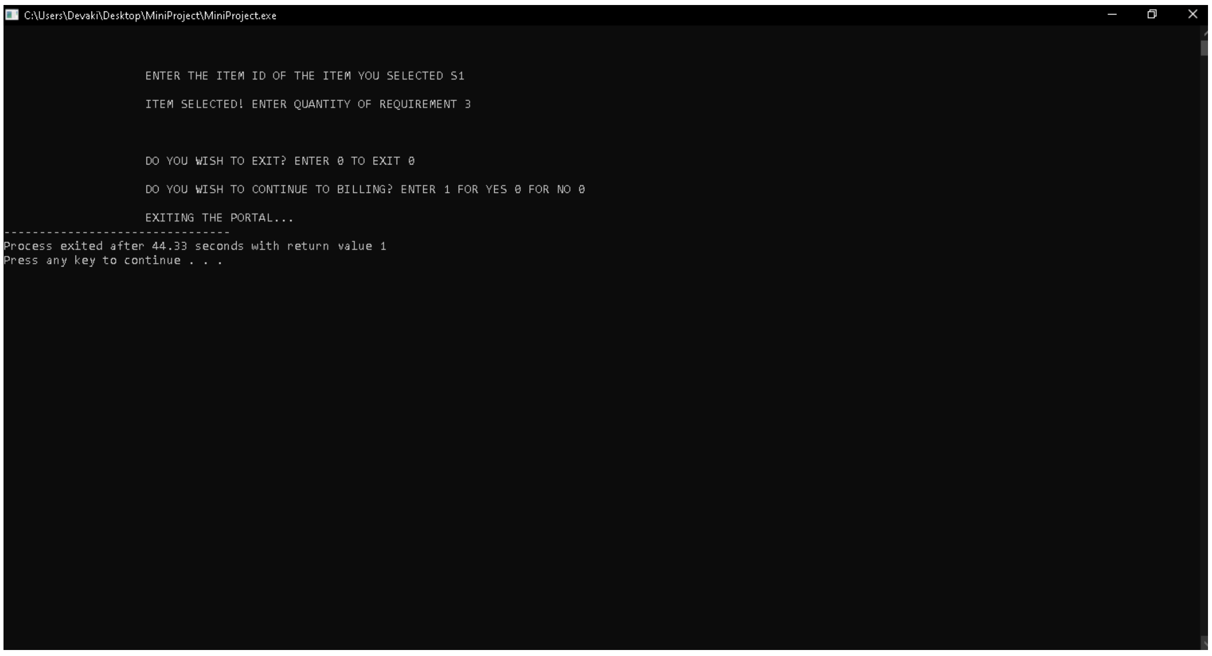
```
C:\Users\Devaki\Desktop\MiniProject\MiniProject.exe

ENTER THE ITEM ID OF THE ITEM YOU SELECTED S2


ITEM NOT FOUND PLEASE TRY AGAIN!

DO YOU WISH TO EXIT? ENTER 0 TO EXIT 1
```

Test Case 9



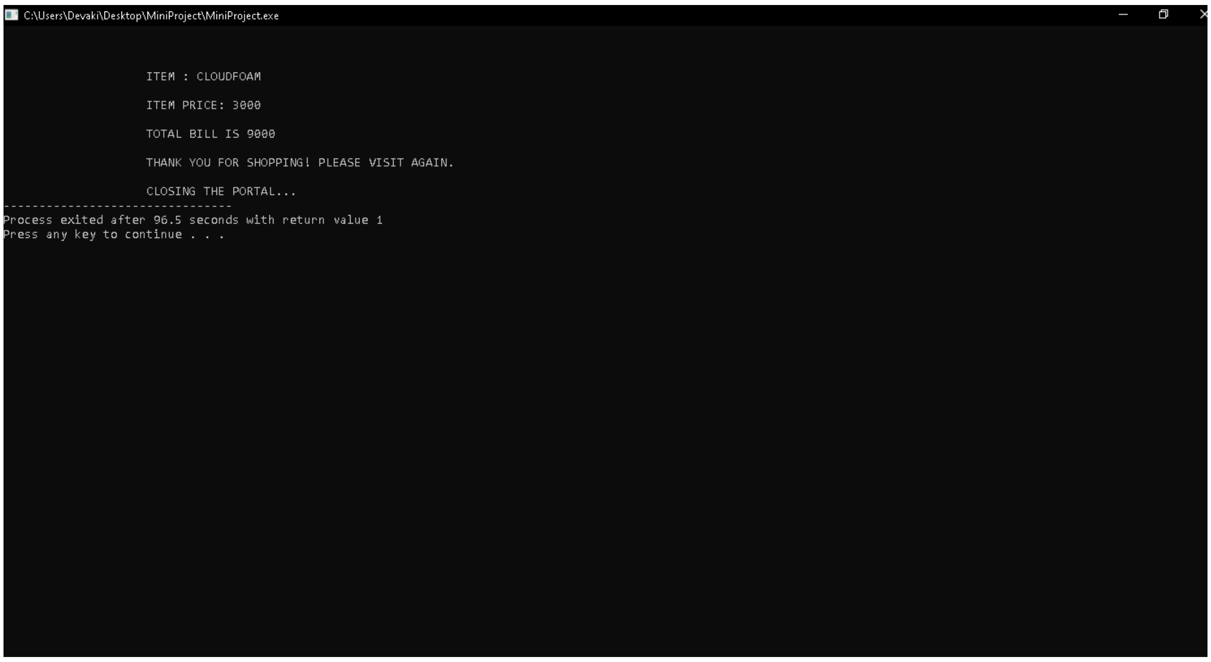
```
C:\Users\Devaki\Desktop\MiniProject\MiniProject.exe

                    ENTER THE ITEM ID OF THE ITEM YOU SELECTED S1

                    ITEM SELECTED! ENTER QUANTITY OF REQUIREMENT 3


                    DO YOU WISH TO EXIT? ENTER 0 TO EXIT 0

                    DO YOU WISH TO CONTINUE TO BILLING? ENTER 1 FOR YES 0 FOR NO 1

                    TAKING YOU TO BILLING...
```

Test Case 10



```
C:\Users\Devaki\Desktop\MiniProject\MiniProject.exe

                    ENTER THE ITEM ID OF THE ITEM YOU SELECTED S1

                    ITEM SELECTED! ENTER QUANTITY OF REQUIREMENT 3


                    DO YOU WISH TO EXIT? ENTER 0 TO EXIT 0

                    DO YOU WISH TO CONTINUE TO BILLING? ENTER 1 FOR YES 0 FOR NO 0

                    EXITING THE PORTAL...
--------------------------------
Process exited after 44.33 seconds with return value 1
Press any key to continue . . .
```

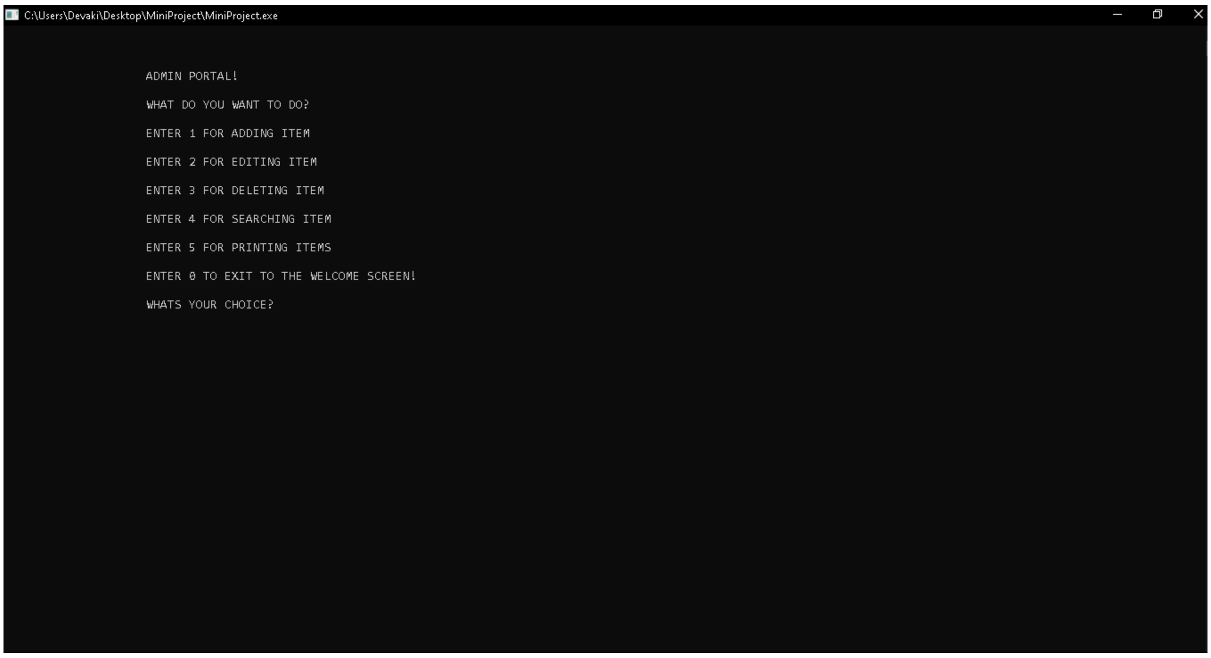
Test Case 11



```
C:\Users\Devaki\Desktop\MiniProject\MiniProject.exe

                    ITEM : CLOUDFOAM

                    ITEM PRICE: 3000

                    TOTAL BILL IS 9000

                    THANK YOU FOR SHOPPING! PLEASE VISIT AGAIN.

                    CLOSING THE PORTAL...
--------------------------------
Process exited after 96.5 seconds with return value 1
Press any key to continue . . .
```

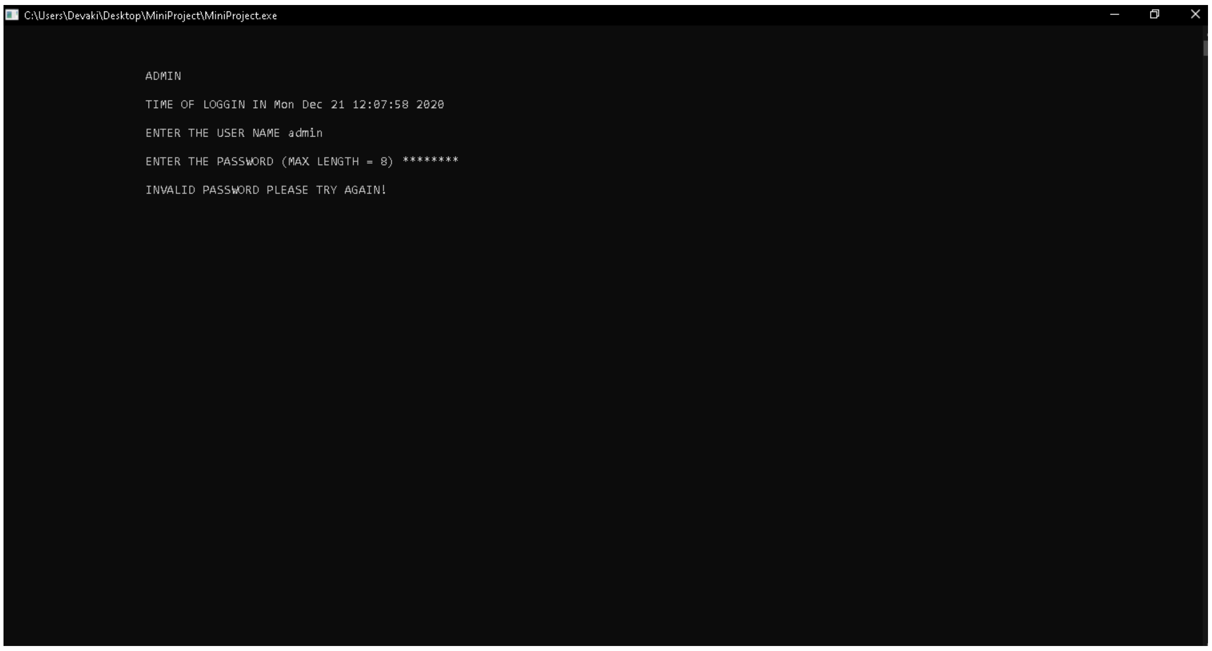
4.2 ADMIN USE CASES

Admin Menu Screen



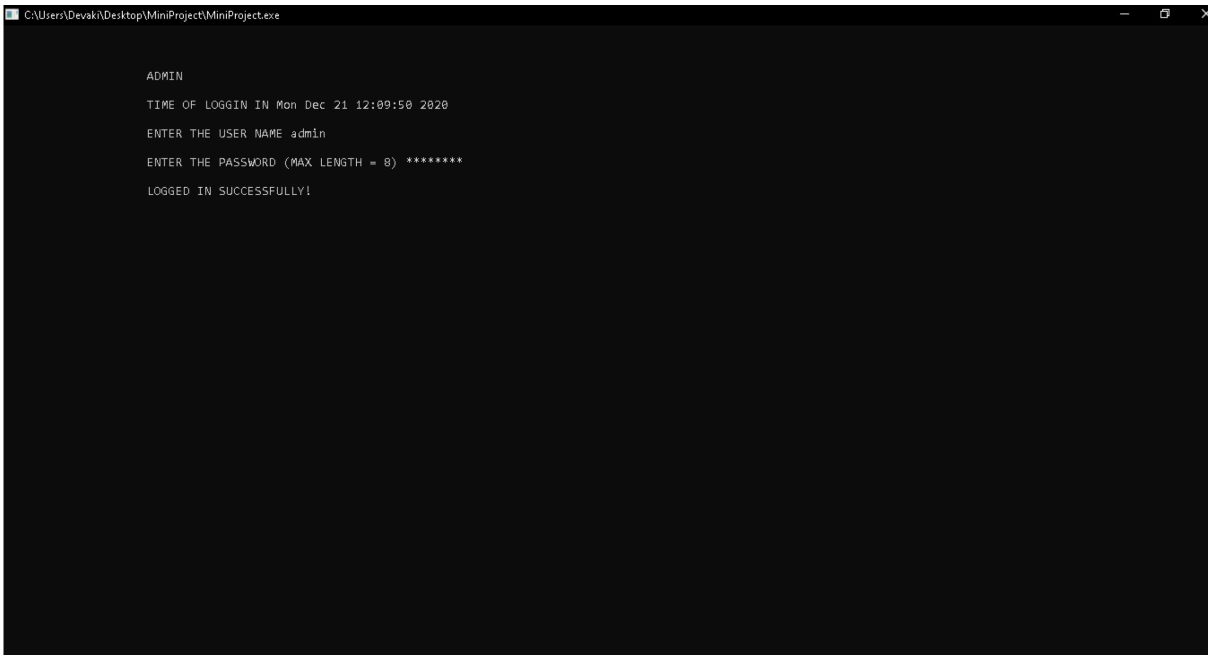
```
C:\Users\Devaki\Desktop\MiniProject\MiniProject.exe

                    ADMIN PORTAL!

                    WHAT DO YOU WANT TO DO?

                    ENTER 1 FOR ADDING ITEM

                    ENTER 2 FOR EDITING ITEM

                    ENTER 3 FOR DELETING ITEM

                    ENTER 4 FOR SEARCHING ITEM

                    ENTER 5 FOR PRINTING ITEMS

                    ENTER 0 TO EXIT TO THE WELCOME SCREEN!

                    WHATS YOUR CHOICE?
```

Test Case 12



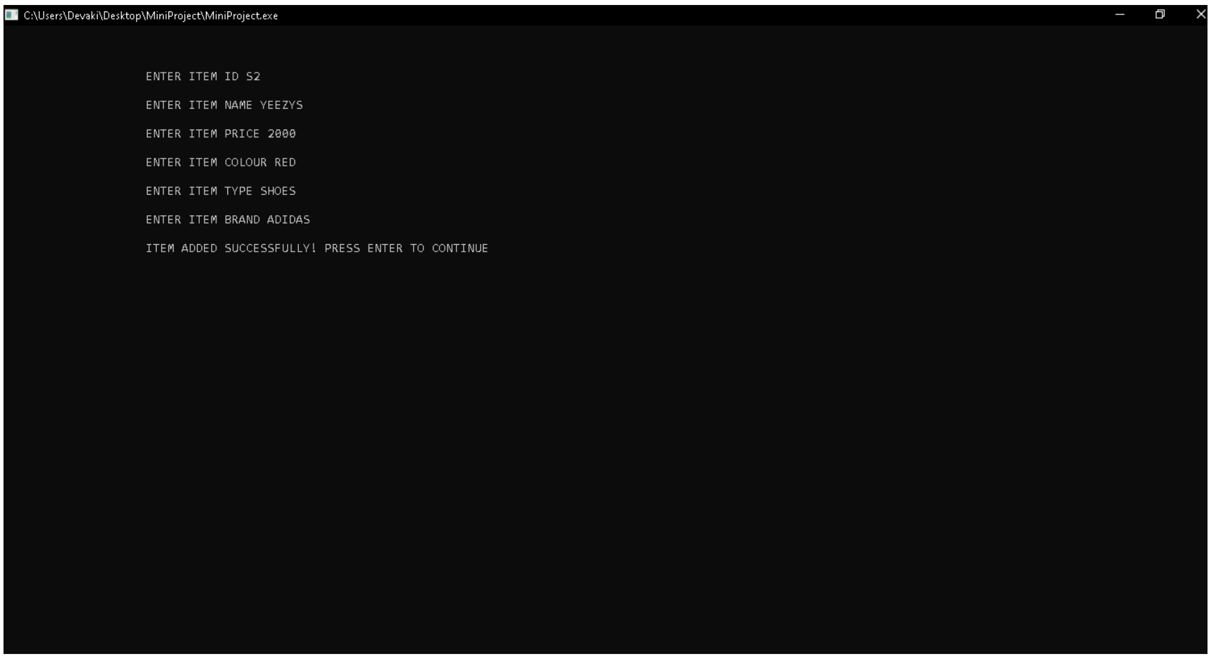
```
C:\Users\Devaki\Desktop\MiniProject\MiniProject.exe

                ADMIN

                TIME OF LOGGIN IN Mon Dec 21 12:07:58 2020

                ENTER THE USER NAME admin

                ENTER THE PASSWORD (MAX LENGTH = 8) ********

                INVALID PASSWORD PLEASE TRY AGAIN!
```

Test Case 13

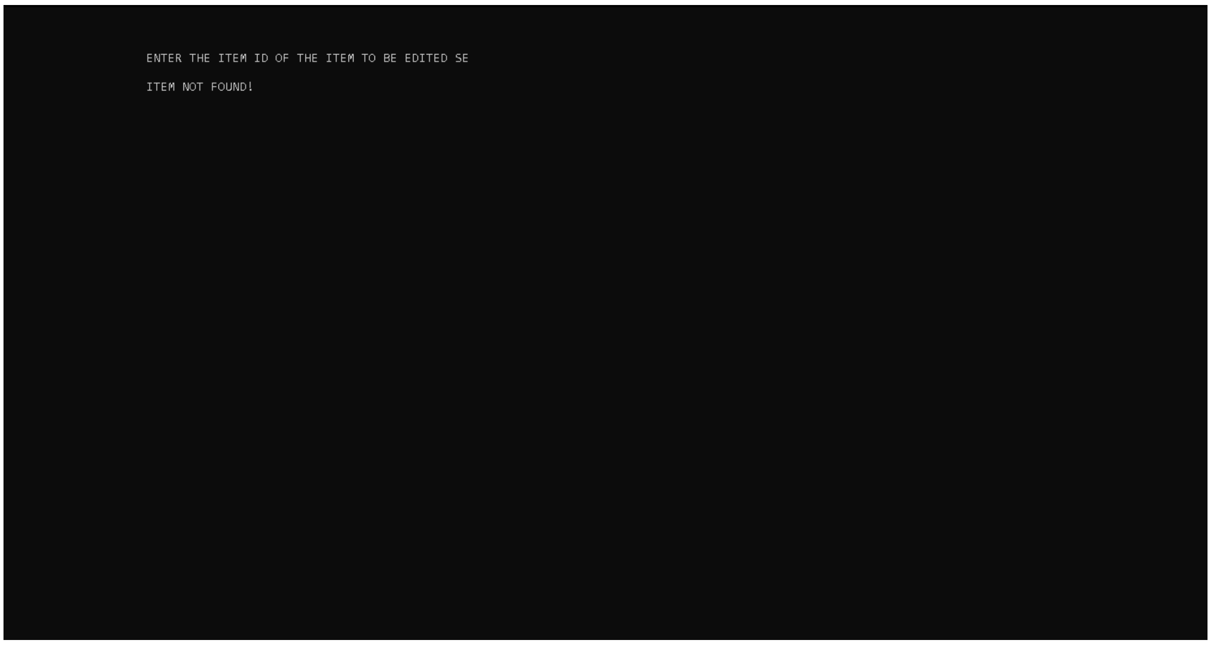


```
C:\Users\Devaki\Desktop\MiniProject\MiniProject.exe

                ADMIN

                TIME OF LOGGIN IN Mon Dec 21 12:09:50 2020

                ENTER THE USER NAME admin

                ENTER THE PASSWORD (MAX LENGTH = 8) ********

                LOGGED IN SUCCESSFULLY!
```

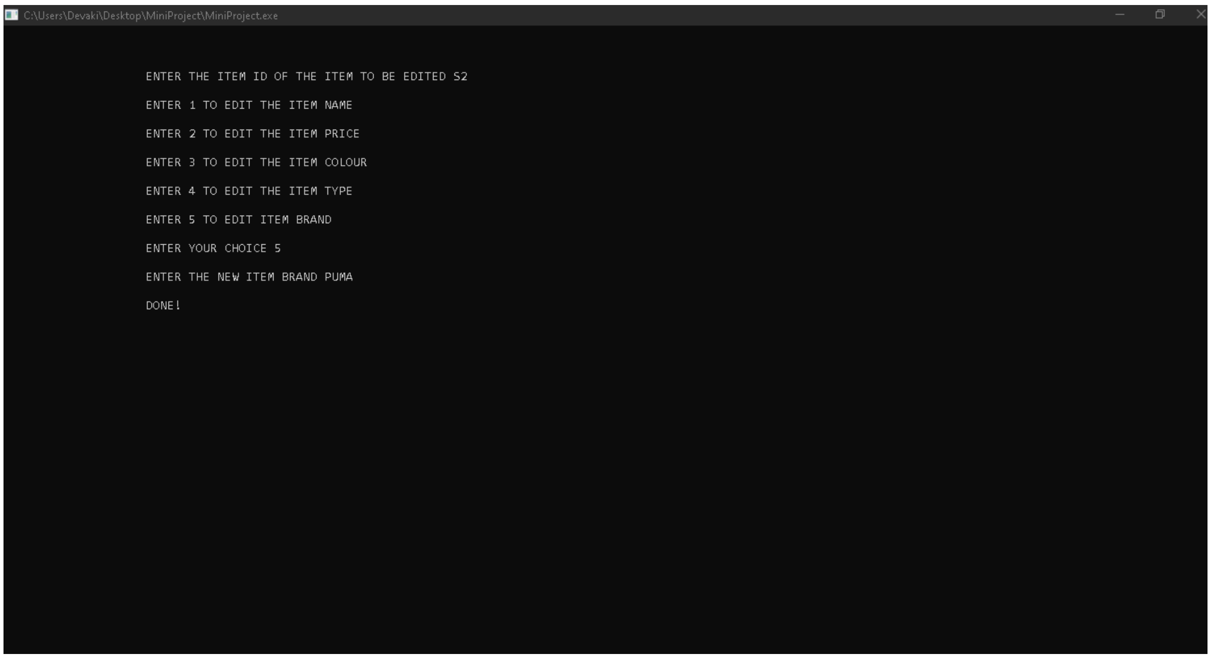
Test Case 14


```
C:\Users\Devaki\Desktop\MiniProject\MiniProject.exe

                    ENTER ITEM ID S2

                    ENTER ITEM NAME YEEZYS

                    ENTER ITEM PRICE 2000

                    ENTER ITEM COLOUR RED

                    ENTER ITEM TYPE SHOES

                    ENTER ITEM BRAND ADIDAS

                    ITEM ADDED SUCCESSFULLY! PRESS ENTER TO CONTINUE
```

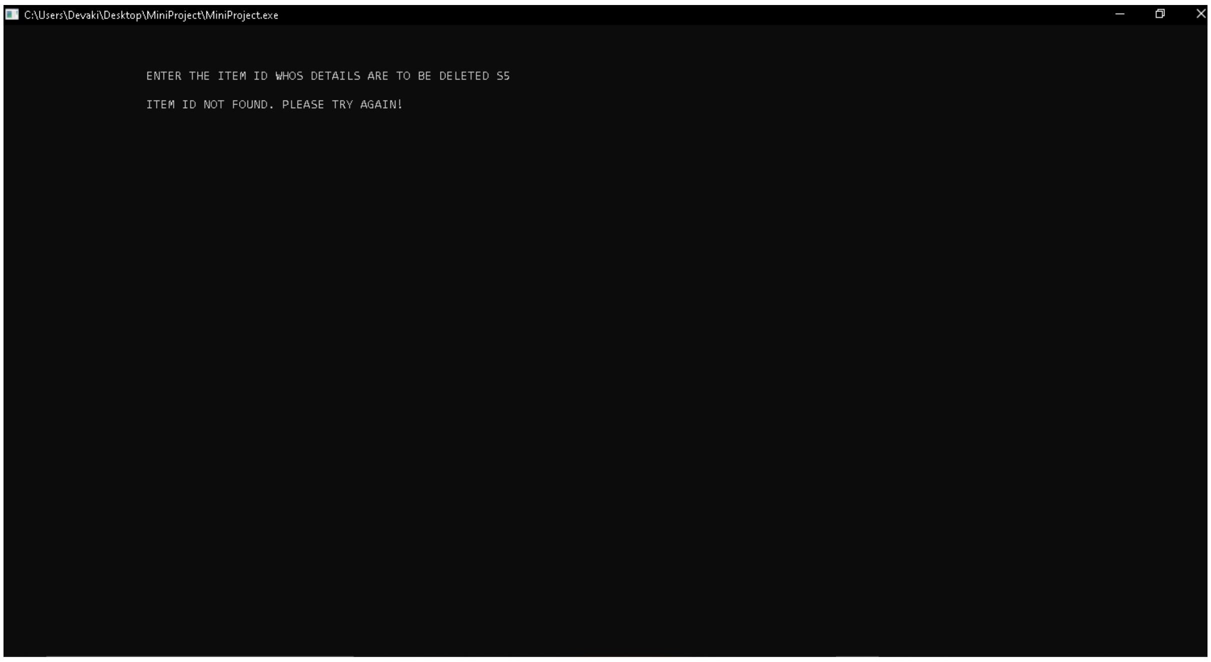
Test Case 15


```
                    ENTER THE ITEM ID OF THE ITEM TO BE EDITED SE

                    ITEM NOT FOUND!
```

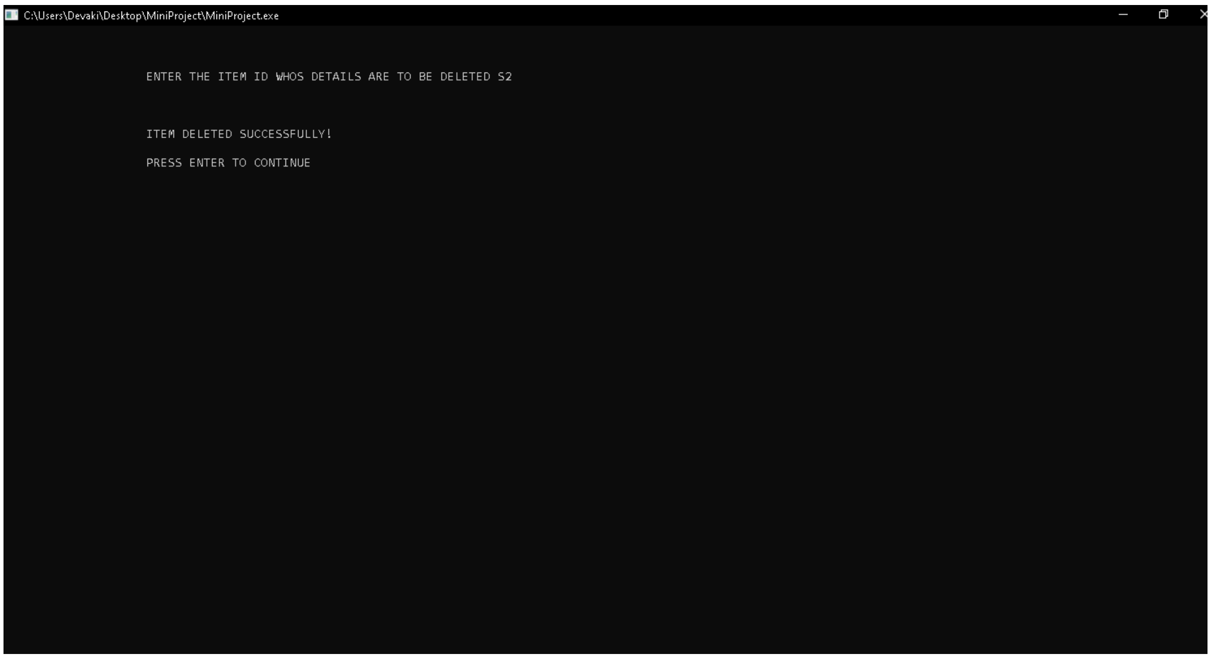
Test Case 16


```
C:\Users\Devaki\Desktop\MiniProject\MiniProject.exe

ENTER THE ITEM ID OF THE ITEM TO BE EDITED S2

ENTER 1 TO EDIT THE ITEM NAME

ENTER 2 TO EDIT THE ITEM PRICE

ENTER 3 TO EDIT THE ITEM COLOUR

ENTER 4 TO EDIT THE ITEM TYPE

ENTER 5 TO EDIT ITEM BRAND

ENTER YOUR CHOICE 5

ENTER THE NEW ITEM BRAND PUMA

DONE!
```

Test Case 17


```
C:\Users\Devaki\Desktop\MiniProject\MiniProject.exe

ENTER THE ITEM ID WHOS DETAILS ARE TO BE DELETED S5

ITEM ID NOT FOUND. PLEASE TRY AGAIN!
```

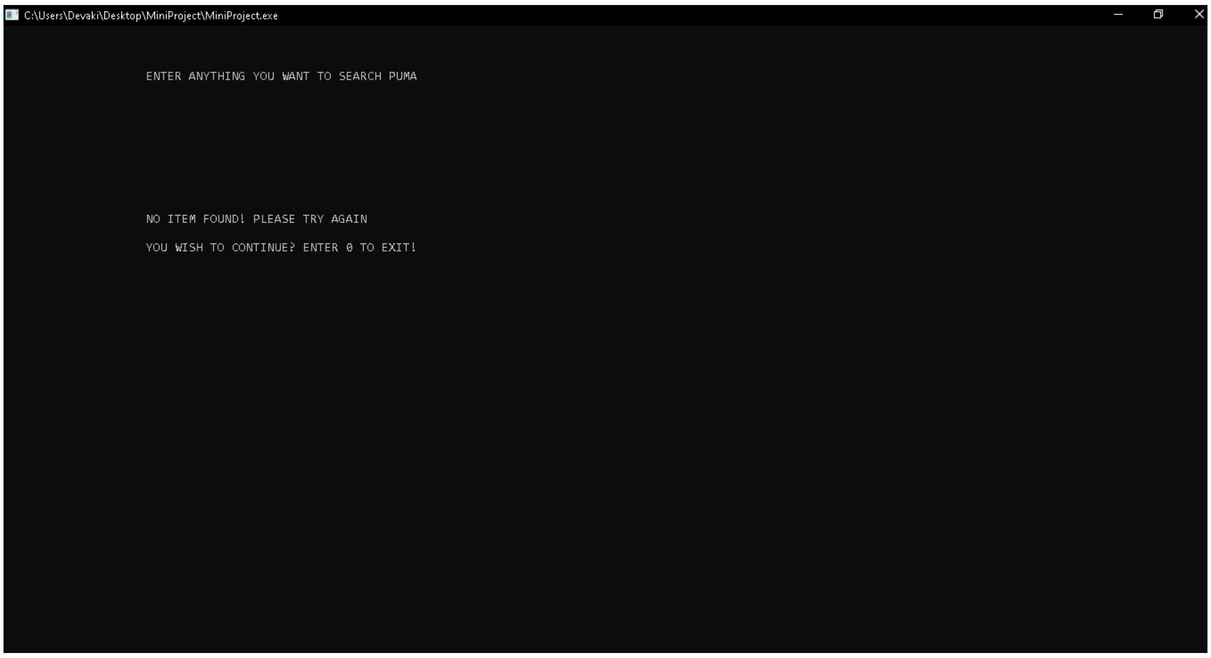
Test Case 18



C:\Users\Devaki\Desktop\MiniProject\MiniProject.exe

```
ENTER THE ITEM ID WHOS DETAILS ARE TO BE DELETED S2


ITEM DELETED SUCCESSFULLY!

PRESS ENTER TO CONTINUE
```

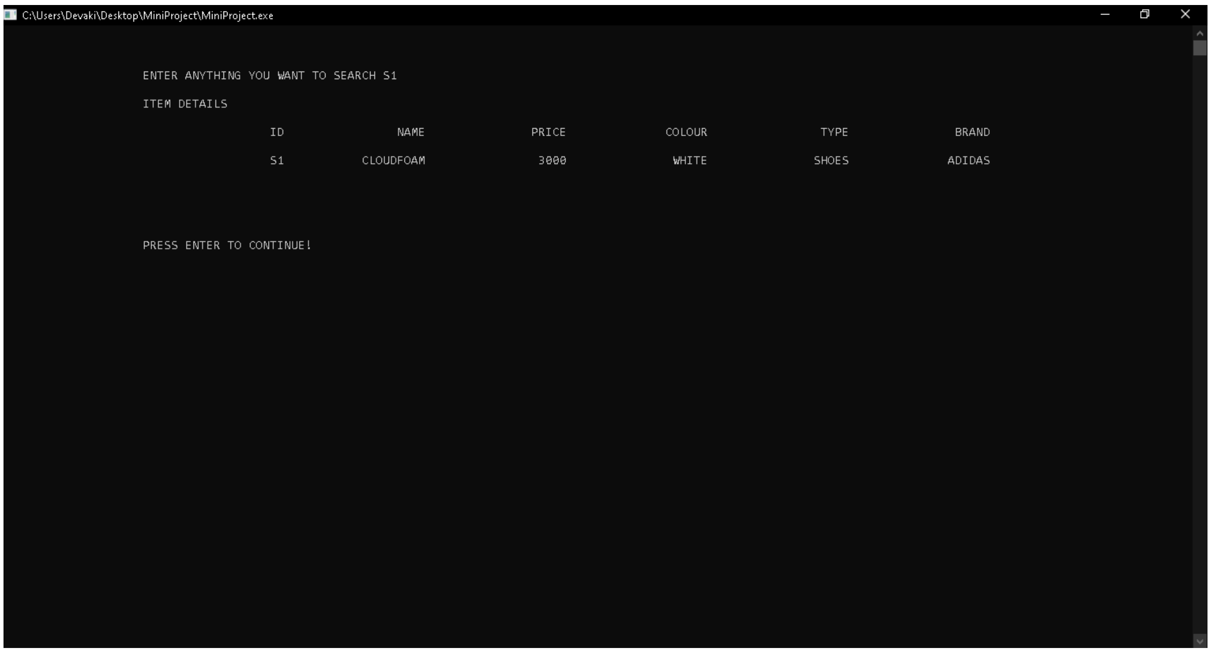
Test Case 19



C:\Users\Devaki\Desktop\MiniProject\MiniProject.exe

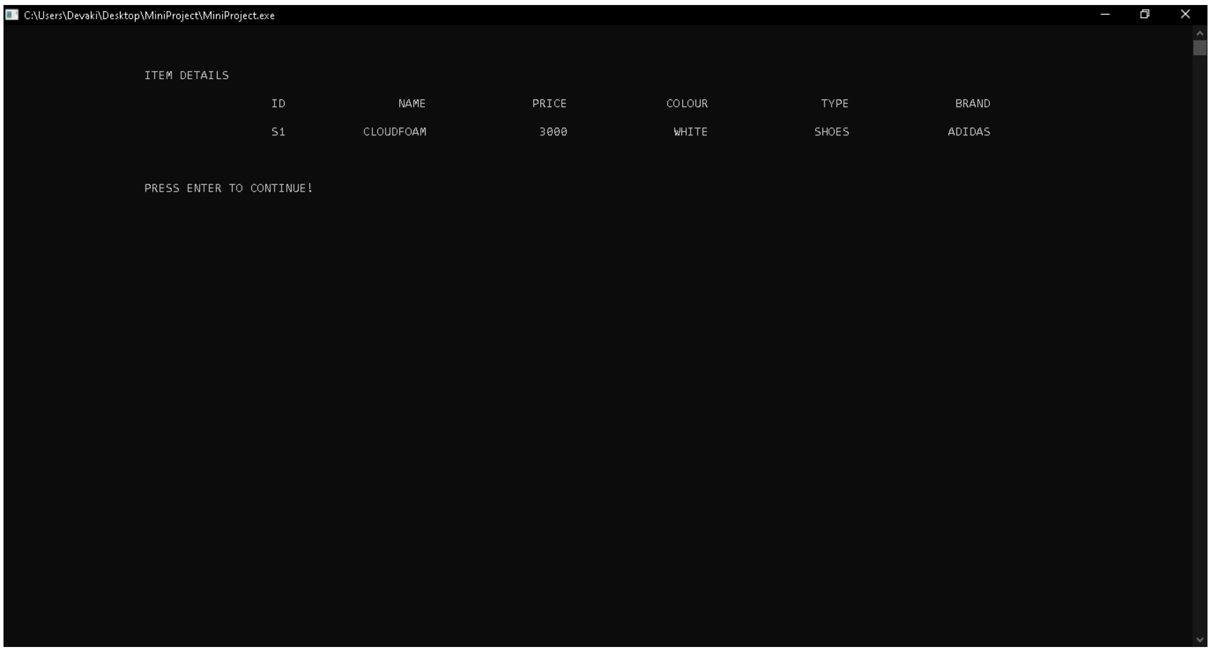
```
ENTER ANYTHING YOU WANT TO SEARCH PUMA




NO ITEM FOUND! PLEASE TRY AGAIN

YOU WISH TO CONTINUE? ENTER 0 TO EXIT!
```

Test Case 20



```
C:\Users\Devaki\Desktop\MiniProject\MiniProject.exe

                ENTER ANYTHING YOU WANT TO SEARCH S1

                ITEM DETAILS

                               ID              NAME              PRICE              COLOUR              TYPE              BRAND

                               S1          CLOUDFOAM              3000               WHITE             SHOES             ADIDAS



                PRESS ENTER TO CONTINUE!
```

Test Case 21



```
C:\Users\Devaki\Desktop\MiniProject\MiniProject.exe

                ITEM DETAILS

                               ID              NAME              PRICE              COLOUR              TYPE              BRAND

                               S1          CLOUDFOAM              3000               WHITE             SHOES             ADIDAS


                PRESS ENTER TO CONTINUE!
```

5. ADDITIONAL KNOWLEDGE ACQUIRED

Implementing this project in C Language has introduced us to different libraries such as: ‘conio.h’, ‘time.h’ and ‘windows.h’. We were able to use the knowledge we have on Structures and File Handing and were able to build this Mini Project. We explored the ‘time.h’ and ‘conio.h’ libraries for achieving a look-and-feel of an actual window application by constructing our own time delay function.

Also, we have further improved in our knowledge in file-handling because of the vast amount of data manipulation we have done using text files.

6. CONCLUSION AND FUTURE WORK

To conclude, we have built a management system which enables vendor to add new items, edit item details, delete items and the customer to select items which they like. The motivation behind this is to digitalize the traditional way of storing and managing data into a much more efficient, fast and safe/protected experience.

Our future work adding a UI and making the experience more seamless, incorporating Data Structures to made the program much faster and even connect to a MySQL database to store and access information much efficiently.

This project can be further improved by converting it into a Web Application using Python and the Django Framework.

7. REFERENCES

1. C Language Documentation: https://docs.microsoft.com/en-us/cpp/clanguage/?view=msvc-160
2. Stack Overflow (for debugging errors): https://stackoverflow.com/
3. Dev C++ (editor): https://sourceforge.net/projects/orwelldevcpp/