**COFFEE HOUSE**

*A*

*Mini Project Report*

*Submitted in partial fulfilment of the*

*Requirements for the award of the Degree of*

**BACHELOR OF ENGINEERING**

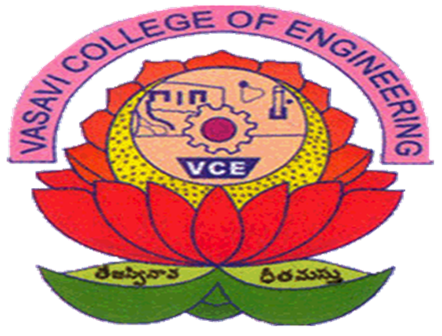
IN

**INFORMATION TECHNOLOGY**

By

**V SHWETHA 1602-19-737-106**

**KVS SRIYA 1602-19-737-115**

****

**Department of Information Technology**

**Vasavi College of Engineering (Autonomous)**

**(Affiliated to Osmania University)**

**Ibrahimbagh, Hyderabad-31**

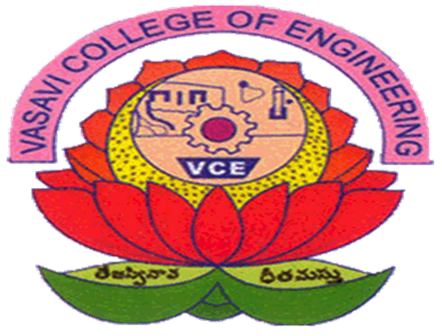
**2020**

**Vasavi College of Engineering (Autonomous)**

**(Affiliated to Osmania University)**

**Hyderabad-500 031**

**Department of Information Technology**

****

**DECLARATION BY THE CANDIDATE**

We, **KVS SRIYA** and **V.SWETHA**, bearing hall ticket numbers, **1602-19-737-115** and **1602-19-737-106** hereby declare that the project report entitled “COFFEE HOUSE” 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 bonafide 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.

**1602-19-737-106 V Shwetha**

**1602-19-737-115 KVS Sriya**

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

**ACKNOWLEDGEMENT**

We would like to thank Ms. PRASANNA for giving us outstanding guidance and co-operation for designing this system. Without her kind patronage and guidance the project would not have taken shape. We would also like to express our gratitude and sincere regards for her kind approval of the project, time to time counselling and advice.

We would also like to thank our GOD for giving this opportunity to build up an application by the use of C Programming.

Next we would also like to thank the entire faculty for their full cooperation.

Last but not the least, we would also like to thank our parents for all their support that continues to motivate us to work towards our betterment and create an environment during this pandemic situation where we could work and complete the project successfully.

So, with gratitude we acknowledge all those who guided and encouraged and served a beacon of light and crowned the effect with success.

**ABSTRACT**

*Coffee house is a place where you can have any refreshments and beverages of your choice.*

*The process of managing the menu ,consumption and payments becomes quite challenging.*

*The main aim of the project is to provide a complete solution for the problems faced in the coffee sales system.*

*It displays the menu, takes orders , displays the bill and maintains a record of all this.*

*This software becomes useful for all the users of coffee houses so that their transactions and details become up to date.*

*This system provides the advantage of having a single server for all purposes .*

**CONTENTS**

|  |  |  |
| --- | --- | --- |
| **S.NO.** | . **CONTENT** | **PAGE NUMBER** |
| **1.0** | **Introduction** | 5 |
| **2.0** | **Technology** | 6 |
| **2.1** | Software requirements | 6 |
| **2.2** | Hardware requirements | 6 |
| **3.0** | **Proposed work** |  |
| **3.1** | **DESIGN** | 8 |
| **3.1.1** | Use case diagram | 8 |
| **3.1.2** | FlowCharts | 9 |
| **3.1.3** | Description of all use cases | 10 |
| **3.2** | **IMPLEMENTATION** | 15 |
| **3.2.1** | Module wise code for the entire project | 15 |
| **3.2.2** | Github links | 27 |
| **3.3** | **TESTING** | 28 |
| **4.0** | **Results** | 34 |
| **5.0** | **Additional knowledge gained** | 62 |
| **6.0** | **Conclusion** | 62 |
| **7.0** | **Future work** | 62 |
| **8.0** | **References** | 63 |

1. **INTRODUCTION**

It is common for Offices, Factories, Call Centres, Hostels, Schools, Clubs and Hospitals to operate their own cafeterias for their employees and students. However, managing the cafeteria menu, billing and consumption is a challenging process. Manual and paper based processes are cumbersome and error-prone, leading to inaccuracies and wastage of time and material. It especially manages menu items , gives interface to the orderings and expenditure report. It is designed to create operational efficiency, saving our money and time, and avoiding paperwork.

Our project “COFFEE HOUSE” offers a software that provides food service to the customers and a password protected admin panel for the owner to plan different menus for different occasions by editing the menu. The project creates a user friendly environment. It increases the efficiency and saves time. The program saves the bills and preferences given by the customer in a file with the date for easier calculation of money inflow into the coffee house.

**FEATURES OF THE PROJECT**

* **Home page**

There will be 3 options provided: customer, admin and exit.

* **Login page**

Admin can login using the username and password. System automatically exits if the username or password is invalid more than four times.

* Food services
  + Select items from menu
  + Confirm order
  + Give preference
  + Response from staff for the preference mentioned by customer
* Display Total bills
* Edit menu
* Edit price
* Edit Quantity
* Add Item
* Delete Item
* Exit

**2. TECHNOLOGY**

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

**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 package and need to be installed separately.

In order to use our Coffee House project, one should have the following software requirements:

* Operating system: windows 7 and above
* C compiler: GNU Compiler Collection (GCC).
* Editor: Any text editor that supports C language

**HARDWARE REQUIREMENTS**

Hardware requirements refer to the common set of 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.

In order to use our Coffee House project, one should have the following hardware requirements:

* Processor: intel core I5 and above
* Memory: 4 GB RAM and above

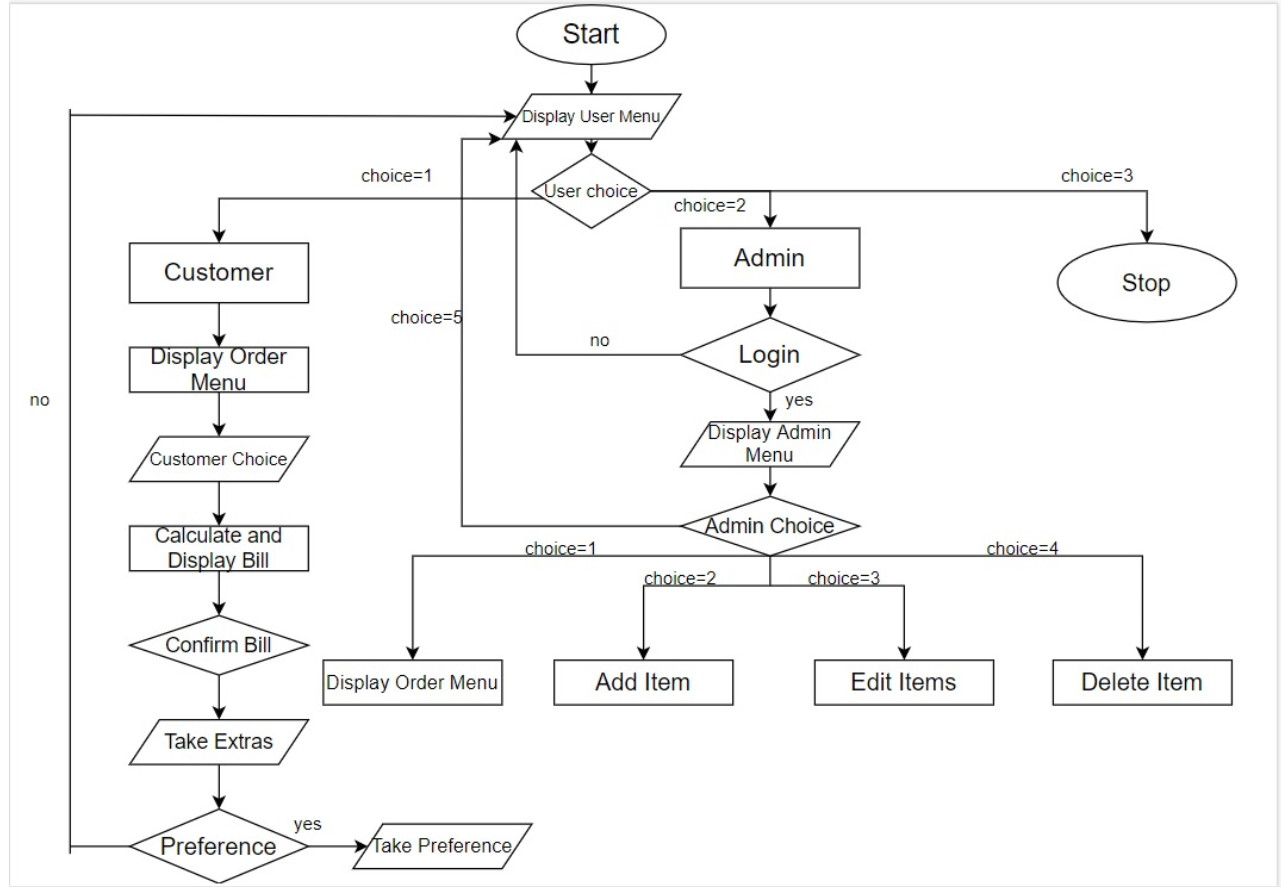
**3. PROPOSED WORK**

**3.1) Design**

**3.1.1) USE CASE DIAGRAM**



**FLOWCHART:**



**USE CASE DESCRIPTIONS**

# Name: Login

Actors: Manager

Pre condition: Manager should register with system

Post condition: Manager logs in and all the conditions are displayed on screen.

|  |  |
| --- | --- |
| **Main flow** | |
| **Actor’s action** | **System response** |
| 1)Enter’s user name and password and chooses login option |  |
|  | 2)Validates user name and password |
|  | 3)Allows access to system |
|  | 4)If types Invalid username or password shows error message |

**Name: Confirm order**

Actors:Customer

Pre condition: Choose the confirm order option from home screen

Pot condition: Customer selects items from menu

|  |  |
| --- | --- |
| **Main Flow** | |
| **Actor’s Action** | **System response** |
| 1)Display menu |  |
|  | 2)Display all the items available in the coffee house |
| 3)Selects the item and quantity of item. |  |
|  | 4)Takes the item and quantity of item . |
| 5)Selects whether to confirm order or delete |  |
|  | 6)If order is confirmed goes to place order  Or Else displays menu again to change order |

# Name: Places order

Actors: Customer

Pre condition: Confirming Order

Post condition: Displays bill

|  |  |
| --- | --- |
| **Main Flow** | |
| **Actor’s Action** | **System’s Response** |
| 1)Selects Place order option |  |
|  | 2)Asks to choose Display bill option. |
| 3)Select Display bill option |  |
|  | 4)Calculates and display’s bill. |
| 5)Choose exit option |  |
|  | 6)Display’s Order is Placed message. |

# Name:Give Preference

# Actors:Customer

# Pre condition:Must have confirmed Order

# Post condition: Sending Notification to Manager

|  |  |
| --- | --- |
| **Main Flow** | |
| **Actor’s Action** | **System’s response** |
| 1)Selects Give Preference Option |  |
|  | 2). Display’s give Preference message |
|  | 3)Take input from actor |
|  | 4)If already in the menu ignores or else sends notification to Manager |

# Name:Recives Order

# Actors: Staff

# Pre condition: Customer Placing Orders

# Post Condition: Prepares The order

|  |  |  |
| --- | --- | --- |
| **Main Flow** | | |
| **Actor’s Action** | **System’s response** | |
| 1)Choosing Receive order option |  | |
|  | 2)Display’s the orders placed by customer’s |

# Name:Prepare and send order

# Actors:staff

# Pre condition: receiving orders

# Post condition: Order delivered

|  |  |
| --- | --- |
| **GENERAL CHARACTERISTICS** | |
| **Actor’s action** | **System’s response** |
| 1)Checking the quantity of order placed |  |
|  | 2)If available display’s deliver the order message |
| 3)Select deliver the order |  |
|  | 4)Display’s order delivered message |

# Name: Edit Menu

# Actors: Manager

# Pre condition:Should be logged in

# Post condition:Can add ,delete item and change price of items

|  |  |
| --- | --- |
| **Main Flow** | |
| **Actor’s action** | **System’s response** |
| 1)Chooses edit menu option |  |
|  | 2)Display’s menu containing 3 option;s   * Add item * Delete item * Change price |
| ***Internal flow:*** |
| **Actor’s action** | **System’s response** |
| 3)Chooses add item option |  |
|  | 4)Takes input from the actor checks if item already there in the menu if not adds into the menu and display’s item added into menu message |
| 5)Chooses delete item option |  |
|  | 6)If item is there in the menu deletes it and display’s deleted item from menu message |
| 7)Choose change the price option |  |
|  | 8)Takes input of item and price from the user and display’s the price of mentioned item has been changed to mentioned price message. |

**Name:Check profit and losses**

Actors:Manager

Pre condition: Must be Logged in

Post condition: Gives profit or loss of the coffee house

|  |  |
| --- | --- |
| **Main flow** | |
| **Actor’s action** | **System’s response** |
| 1)Choose to check the profit or loss option |  |
|  | 2)Display’s the record of payment’s by customers |
|  | 3)Calculates the total amount of money and display’s the total amont of inflow that day with date. |

* 1. **Implementation**

**3.2.1) Module wise code for the entire project**

/\*

COFFEE HOUSE IN C

V SWETHA 1602-19-737-106

KVS SRIYA 1602-19-737-115

\*/

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

#include<ctype.h>

#include<time.h>

int order\_count=0;

struct item

{

int sno;

float price;

char item\_name[50];

int quantity;

char ingredients[1000];

};

struct item list[100];

int num=1;

float total\_bill;

int login()

{

int type=0,i;

char username[10],ch=' ';

char password[10];

char user[10]="coffee";

char pass[10]="house";

do

{

printf(" \n\n ENTER USERNAME:-");

scanf("%s", &username);

printf(" \n\n ENTER PASSWORD:-");

i=0;

while(i<10)//for password not to appear on screen

{

password[i]=getch();

ch=password[i];

if(ch==13) break;

else printf("\*");//\* appears when we type password

i++;

}

password[i]='\0';

i=0;

if(strcmp(username,"coffee")==0 && strcmp(password,"house")==0)

{

printf(" \n\n\n LOGIN IS SUCCESSFUL!!");

return 1;

break;

}

else

{

printf("\n SORRY, LOGIN IS UNSUCESSFUL!!\nEnter again");

type++;

getch();//holds the screen

}

}

while(type<=2);

if (type>2)

{

printf("\nSorry you have entered the wrong username and password for four times!!!");

return 2;

}

system("cls");//It clears the output screen of the program

}

void insert(char name[50],float rate,int number,char lists[1000])

{

list[num].sno=num;

list[num].price=rate;

strcpy(list[num].item\_name,name);

strcpy(list[num].ingredients,lists);

list[num].quantity=number;

num++;

}

void edit\_name(int sno,char name[50])

{

strcpy(list[sno].item\_name,name);

return ;

}

void edit\_price(int sno,float rate)

{

list[sno].price=rate;

return ;

}

void edit\_quantity(int sno, int number)

{

list[sno].quantity=number;

return ;

}

void display()

{

system("cls");

printf("\n\n\n\n\n\n\t\t\tTHE COFFEE HOUSE");

printf("\n\n TODAYS MENU...");

int i=1;

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

printf("| SNO. | NAME | PRICE | QUANTITY |\n");

for(i=1;i<num;i++)

{

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

printf("| %d |",i);

//char temp[50];

//temp=list[i].item\_name;

int j=0;

j=20-(strlen(list[i].item\_name));

printf(" %s",list[i].item\_name);

for(j;j<=20;j++)

printf(" ");

printf(" | %0.2f |",list[i].price);

printf(" %d |\n",list[i].quantity);

}

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

}

void delete(int sno)

{

int i=1;

for(i=1;i<num;i++)

{

if(i==sno||i>sno)

{

list[i+1].sno=i;

list[i]=list[i+1];

}

}

num--;

}

void admin()

{

int result=login();

if(result==2)

return ;

int ch1,ch2;

//system("cls");

ch1=ch2=0;

int number;

int edit;

float rate=0.0;

char name[50];

char lists[1000];

while(ch1!=5)

{

start:system("cls");

printf("\n\n\n\n\n\n\t\t\tTHE COFFEE HOUSE");

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

printf("1.Display Menu\n");

printf("\t\t\t2.Add Item\n");

printf("\t\t\t3.Edit menu\n");

printf("\t\t\t4.Delete Item\n");

printf("\t\t\t5.Go back....\n");

printf("\t\t\tEnter your choice:");

scanf("%d",&ch1);

switch(ch1)

{

case 1:display();

printf("\npress 1 to go back");

int key;

scanf("%d",&key);

break;

case 2:system("cls");

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

display();

printf("Enter the name of the coffee: ");

scanf("%s",name);

printf("\tEnter the Quantity available");

scanf("%d",&number);

printf("\tEnter the price of the coffee:");

scanf("%f",&rate);

printf("\tEnter the ingredients");

scanf("%[^\n]",lists);

insert(name,rate,number,lists);

goto start;

case 3: start1:system("cls");

printf("\n\n\n\n\n\n\t\tTHE COFFEE HOUSE");

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

display();

printf("1.Edit name\n");

printf("\t2.Edit price\n");

printf("\t3.Edit quantity\n");

printf("\t4.goback");

scanf("%d",&ch2);

if(ch2==4)

goto start;

else if(ch2==1)

{

printf("Enter the sno. of coffee:");

scanf("%d",&edit);

printf("Enter the new name of the coffee:");

scanf("%s",name);

edit\_name(edit,name);

}

else if(ch2==2)

{

printf("Enter the sno. of coffee:");

scanf("%d",&edit);

printf("Enter the new price of coffee:");

scanf("%f",&rate);

edit\_price(edit,rate);

}

else if(ch2==3)

{

printf("Enter the sno. of coffee:");

scanf("%d",&edit);

printf("Enter the new Quantity of coffee:");

scanf("%d",&number);

edit\_quantity(edit,number);

}

else

{

printf("please enter valid choice");

goto start1;

}

break;

case 4:display();

printf("enter the sno. to delete:");

scanf("%d",&edit);

delete(edit);

break;

case 5:break;

default:printf("please enter valid option");

}

}

return ;

}

float calculate\_bill(int number,int quant)

{

float rate\_of\_item=list[number].price;

return (rate\_of\_item\*quant);

}

void reduce\_quantity(int a[][2],int temp)

{

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

{

list[a[i][0]].quantity=list[a[i][0]].quantity-a[i][1];

}

return ;

}

void backup(float bill)

{

char date[30]=\_\_DATE\_\_;

FILE \*bills;

time\_t now;

struct tm \*time=localtime(&now);

strcat(date,".txt");

bills=fopen(date,"a");

fprintf(bills,"Bill:%0.2f\n",bill);

fclose(bills);

}

void preference()

{

char temp;

char extra[500];

printf("ANY THING ELSE SIR/MADAM:");

scanf("%c",&temp);

scanf("%[^\n]",extra);

//printf("%s",extra);

printf("\t\tWOULD YOU LIKE US TO ADD SOMETHING IN THE MENU\n\t\tIF YES PLEASE ENTER THE NAME OF THE ITEM ELSE ENTER NO: ");

char prefer[200];

scanf("%c",&temp);

scanf("%[^\n]",prefer);

char date[30]=\_\_DATE\_\_;

FILE \*PREFER;

time\_t now;

struct tm \*time=localtime(&now);

strcat(date,".\_preferences");

strcat(date,".txt");

PREFER=fopen(date,"a");

char ch=getc(PREFER);

while(ch!=EOF)

{

ch=getc(PREFER);

}

fprintf(PREFER,"Order number: %d\t",order\_count);

fprintf(PREFER,"Extras: %s",extra);

if(strcmp("no",prefer)!=0)

{

char answer[3];

printf("\n\t THE ITEM ENTERED BY THE CUSTOMER IN THE MENU OR NOT:");

scanf("%s",answer);

fprintf(PREFER,"\t\tPreference: %s\n",prefer);

fclose(PREFER);

}

system("pause");

}

int serve ()

{

printf ("\n\n\t\tORDER NO. %d IS READY\t\t", order\_count);

printf ("\n\n\t\tENJOY YOUR COFFEE......\n\t\tTHANK YOU\n\t\tPLEASE VISIT AGAIN.......!!\n\n");

}

void customer()

{

int ch,number;

ch=0;

int n,a[10][2],temp;

int quantity;

float bill=0.0;

system("cls");

printf("\n\n\n\n\n\n\t\t\tWELCOME TO THE COFFEE HOUSE");

printf("\n\n\n\n\t\t1.See the menu\n\t\t2.Go back\n\t\t");

scanf("%d",&ch);

if(ch==1)

{

int choice=0;

while(choice!=1)

{

display();

printf("\n\n\tPLEASE ENTER NUMBER OF ITEMS REQUIRED:");

scanf("%d",&n);

temp=n;

int i=0;

int a[n][2];

while(n!=0)

{

printf("\n\n\tPLEASE ENTER YOUR CHOICE: ");

scanf("%d",&number);

printf("\t\t%s",list[number].ingredients);

a[i][0]=number;

printf("\n\tENTER THE QUANTITY REQUIRED: ");

scanf("%d",&quantity);

a[i][1]=quantity;

bill=bill+calculate\_bill(number,quantity);

i++;

n--;

}

printf("\n\tYOUR TOTAL BILL IS: %0.2f",bill);

printf("\n\n\tPLEASE ENTER 1 TO CONFIRM YOUR BILL\n\t\t\t2 TO GO BACK TO MENU\t\t");

scanf("%d",&choice);

if(choice==1)

{

order\_count++;

printf("\n\n\t\tYOUR ORDER NUMBER IS: %d \n\t\tYOU WILL SOON RECIEVE YOUR ORDER PLEASE WAIT........",order\_count);

reduce\_quantity(a,temp);

backup(bill);

total\_bill=total\_bill+bill;

preference();

serve();

return ;

}

}

}

return ;

}

int main()

{

system("cls");

system("color B0");

FILE \*end;

int ch;

ch=0;

insert(" Americano ",30.0,12,"Expresso and Slow brew");

insert(" Cappuchino ",50.0,15,"Espresso, steamed milk");

insert(" Cafe Latte ",45.0,15,"Espresso,steamed milk, foam");

insert(" CafeChoco ",50.0,15,"creamy chocloates, steamed milk");

insert(" Cafe Mocha ",40.0,15,"Espresso,steamed milk, foam, creamy chocolate");

insert(" House Tea ",60.0,15,"Spicy nutty indian Tea");

insert("VanillaLatte",70.0,15,"Espresso,foam,sweet Vanilla");

insert(" Frappucino ",80.0,15,"frozen blended espresso");

while(ch!=3)

{

system("cls");

printf("\n\n\n\n\n\n\t\t\tWELCOME TO THE COFFEE HOUSE");

printf("\n\n\n\n\n\n\t\t1.CUSTOMER\n\t\t2.ADMIN\n\t\t3.EXIT\n\t\t");

scanf("%d",&ch);

if(ch==1)

{

customer();

}

else if(ch==2)

{

admin();

}

else if(ch==3)

{

break;

}

else

{

printf("\n\t\t\Enter valid option");

}

}

char date[30]=\_\_DATE\_\_;

time\_t now;

struct tm \*time=localtime(&now);

strcat(date,".txt");

end=fopen(date,"a");

fprintf(end,"\n\n\t\t\t\t\t\tTotal Bill:%0.2f\n",total\_bill);

fclose(end);

return 0;

}

**3.2.3) Github links**

*KVS Sriya :* *https://github.com/Sriya2020/sriyaKVS.git*

*V Shwetha:* *https://github.com/VanguruShwetha*

* 1. **TESTING – TEST CASES**

**ADMIN:**

* + - 1. **Login**

|  |  |  |
| --- | --- | --- |
| **Test Case Template** | | |
| **Test Case ID:** TC01 | | **User Case ID:**  UC01 |
| **Test Case Title:** Login | |
| **Test Case Description:** Administrator attempts to login into the application with incorrect credentials for four times | |
| **Test Steps:** | **Expected Result:** | **Actual Result:** |
| System displays prompt for user to enter username  And password(predefined) | System should display “Sorry Login is unsuccessful!!”  Enter again. | .“Sorry Login is unsuccessful!!”  Enter again. Is displayed. |

|  |  |  |
| --- | --- | --- |
| **Test Case Template** | | |
| **Test Case ID:** TC02 | | **User Case ID:**  UC02 |
| **Test Case Title:**Login | |
| **Test Case Description:** Administrator attempts to login into the application with the correct credentials | |
| **Test Steps:** | **Expected Result:** | **Actual Result:** |
| System displays prompt for user to enter username and password | System should Enter into the environment of the application. | Enters into the application. |

* + - 1. **ADD ITEM**

|  |  |  |
| --- | --- | --- |
| **Test Case Template** | | |
| **Test Case ID: TC03** | | **User Case ID:**  UC03 |
| **Test Case Title: Add Item** | |
| **Test Case Description: Administrator add the details of coffee** | |
| **Test Steps:** | **Expected Result:** | **Actual Result:** |
| System displays the menu and asks to enter the details | System takes the details successfully. | System takes the details successfully. |

* + - 1. **EDIT MENU**

|  |  |  |
| --- | --- | --- |
| **Test Case Template** | | |
| **Test Case ID: TC04** | | **User Case ID:**  UC04 |
| **Test Case Title: Edit Menu** | |
| **Test Case Description: Administrator edits the details of coffee in the menu** | |
| **Test Steps:** | **Expected Result:** | **Actual Result:** |
| System displays the menu. | System displays to edit price, edit name, edit quantity and go back. | System displays to edit price, edit name, edit quantity and go back. |

* + - 1. **EDIT NAME**

|  |  |  |
| --- | --- | --- |
| **Test Case Template** | | |
| **Test Case ID: TC05** | | **User Case ID:**  UC05 |
| **Test Case Title: Edit Name** | |
| **Test Case Description: Administrator edits the name of the coffee in the menu and implement the changes in the menu.** | |
| **Test Steps:** | **Expected Result:** | **Actual Result:** |
| System displays to enter the serial number and new name of the coffee. | System should implement the respective changes and goes back to the main menu. | System implements the respective changes and goes back to the main menu. |

* + - 1. **EDIT PRICE**

|  |  |  |
| --- | --- | --- |
| **Test Case Template** | | |
| **Test Case ID: TC06** | | **User Case ID:**  UC06 |
| **Test Case Title: Edit Price** | |
| **Test Case Description: Administrator edits the price of the coffee in the menu and implement the changes in the menu.** | |
| **Test Steps:** | **Expected Result:** | **Actual Result:** |
| System displays to enter the serial number and new price of the coffee. | System should implement the respective changes and goes back to the main menu. | System implements the respective changes and goes back to the main menu. |

* + - 1. **EDIT QUANTITY**

|  |  |  |
| --- | --- | --- |
| **Test Case Template** | | |
| **Test Case ID: TC07** | | **User Case ID:**  UC07 |
| **Test Case Title: Edit Quantity** | |
| **Test Case Description: Administrator edits the quantity available of the coffee in the menu and implement the changes in the menu.** | |
| **Test Steps:** | **Expected Result:** | **Actual Result:** |
| System displays to enter the serial number and new quantity of the coffee. | System should implement the respective changes and goes back to the main menu. | System implements the respective changes and goes back to the main menu. |

* + - 1. **DELETE ITEM**

|  |  |  |
| --- | --- | --- |
| **Test Case Template** | | |
| **Test Case ID: TC08** | | **User Case ID:**  UC08 |
| **Test Case Title: Delete Item** | |
| **Test Case Description: Administrator delete the details of coffee and changes the serial numbers in the menu.** | |
| **Test Steps:** | **Expected Result:** | **Actual Result:** |
| System displays the menu and asks to enter the serial number to delete the records. | System should implement the respective changes and goes back to the main menu | System implements the respective changes and goes back to the main menu. |

* + - 1. **DISPLAY MENU**

|  |  |  |
| --- | --- | --- |
| **Test Case Template** | | |
| **Test Case ID: TC09** | | **User Case ID:**  UC09 |
| **Test Case Title: Display menu** | |
| **Test Case Description: Displays the menu after modifying the details** | |
| **Test Steps:** | **Expected Result:** | **Actual Result:** |
| System displays the menu and asks to enter 1 to go back to admin panel. | System should implement the respective changes done by adding, edit and deleting items and display the menu | System implements the respective changes done by adding, edit and deleting items and display the menu |

**CUSTOMER:**

* + - 1. **SEE THE MENU**

|  |  |  |
| --- | --- | --- |
| **Test Case Template** | | |
| **Test Case ID: TC10** | | **User Case ID:**  UC10 |
| **Test Case Title: See the menu** | |
| **Test Case Description: Displays todays menu** | |
| **Test Steps:** | **Expected Result:** | **Actual Result:** |
| System displays the menu and asks to enter the details to place the order. | System displays the menu and asks to enter the details to place the order | System displays the menu and asks to enter the details to place the order |

|  |  |  |
| --- | --- | --- |
| **Test Case Template** | | |
| **Test Case ID: TC11** | | **User Case ID:**  UC11 |
| **Test Case Title: See the menu** | |
| **Test Case Description: Displays todays menu and takes the quantity and required items from customer** | |
| **Test Steps:** | **Expected Result:** | **Actual Result:** |
| System displays the menu and asks to enter the number of items required and to enter the choice to place the order. | The system should display the ingredients of the coffee when the user enters the choice. | System displays the ingredients of the coffee when the user enters the choice. |

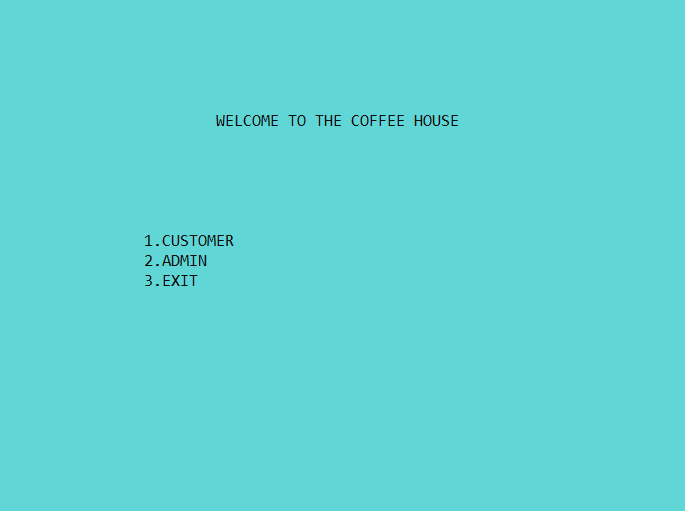
|  |  |  |
| --- | --- | --- |
| **Test Case Template** | | |
| **Test Case ID: TC12** | | **User Case ID:**  UC12 |
| **Test Case Title: To confirm the bill** | |
| **Test Case Description: Displays total bill made by the customer and confirms the order from them.** | |
| **Test Steps:** | **Expected Result:** | **Actual Result:** |
| System displays the total bill and asks to enter 1 to confirm and 2 to go back to menu. | The system should display the order number and a message to wait for your order | System displays the order number and a message saying you will soon receive your order and asks for preference from the customer. |

|  |  |  |
| --- | --- | --- |
| **Test Case Template** | | |
| **Test Case ID: TC13** | | **User Case ID:**  UC13 |
| **Test Case Title: Preference from customer** | |
| **Test Case Description: The customer enters the preferred item.** | |
| **Test Steps:** | **Expected Result:** | **Actual Result:** |
| System asks to enter the preferred item if any else enter any number to confirm the order | Scans the item entered by customer and asks the user to enter if the item is there in the menu scans yes or no and displays to the customer. | Scans the item entered by customer and asks the user to enter if the item is there in the menu scans yes or no and displays to the customer and asks to enter any number to go back. |

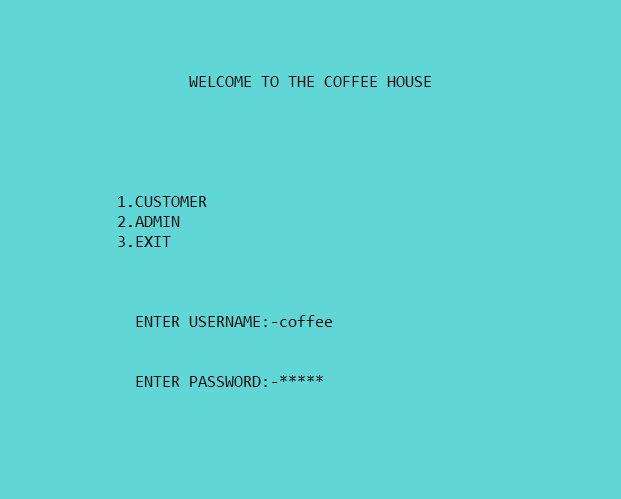
**4.0)** **RESULTS**

**SCREENSHOTS OF EXECUTION**

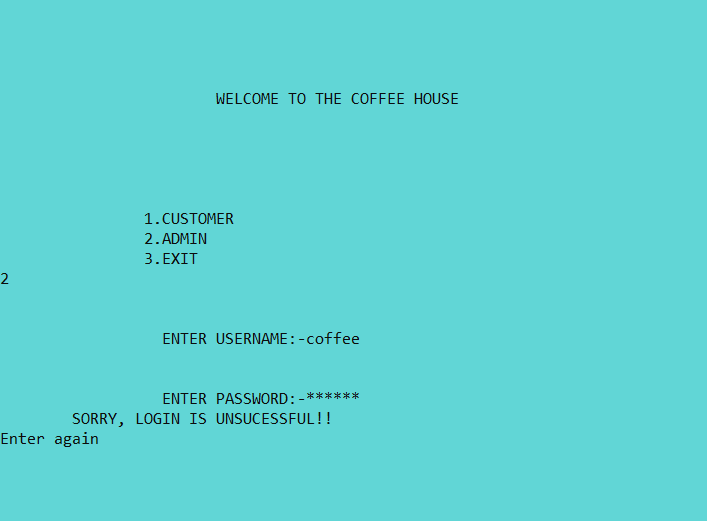
**Start page**



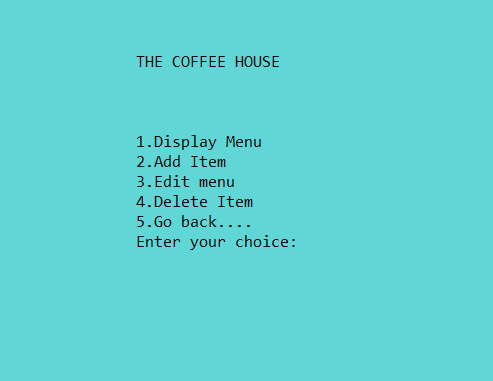
**If user enters 2**



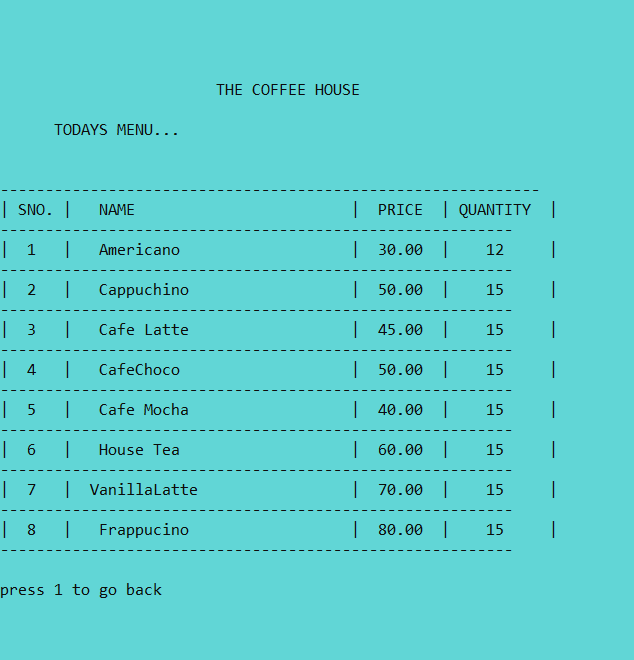
**If username or password is invalid**



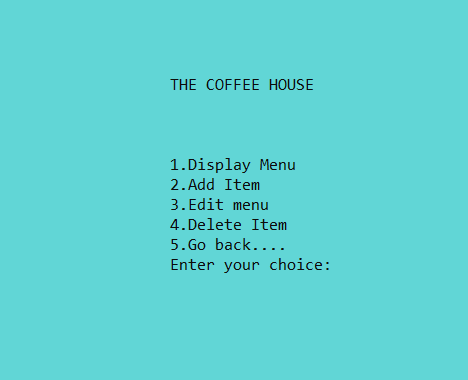
**When user enters correct password and username**



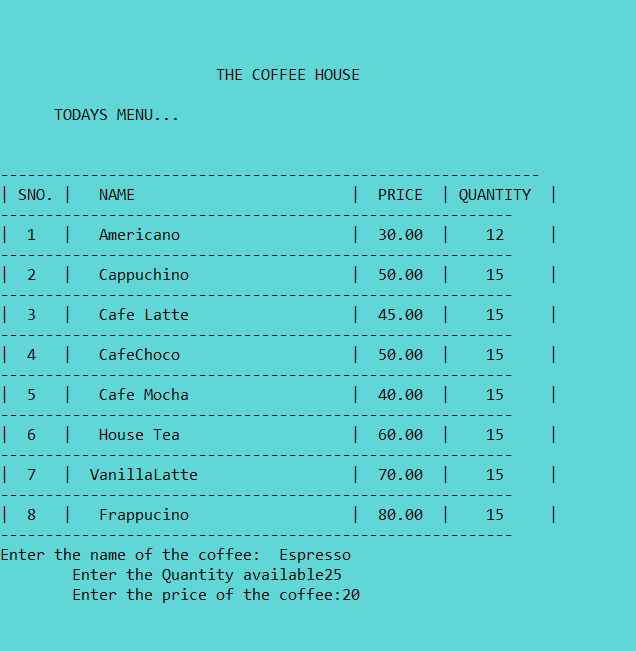
**If you enter 1**



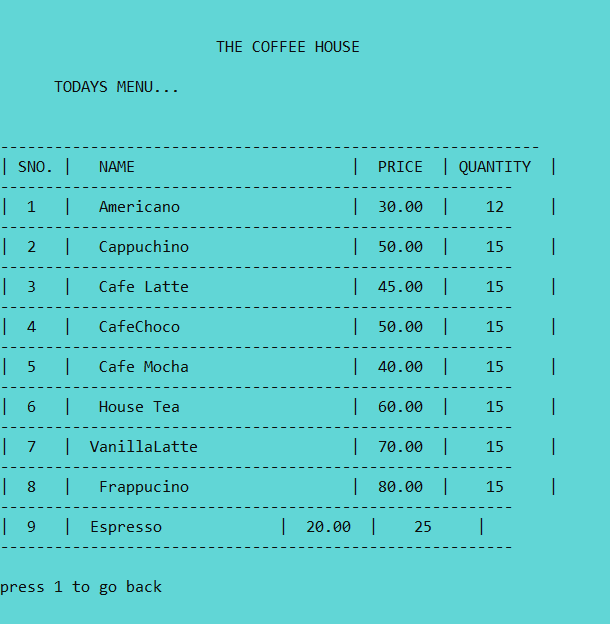
**Again enter 1**

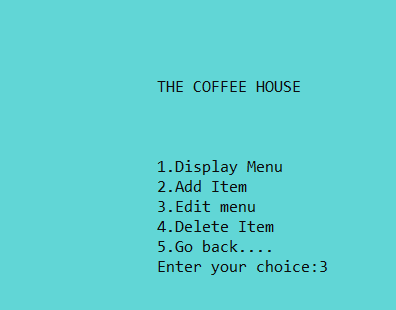


**If you enter 2**



**Now select display menu(1)**

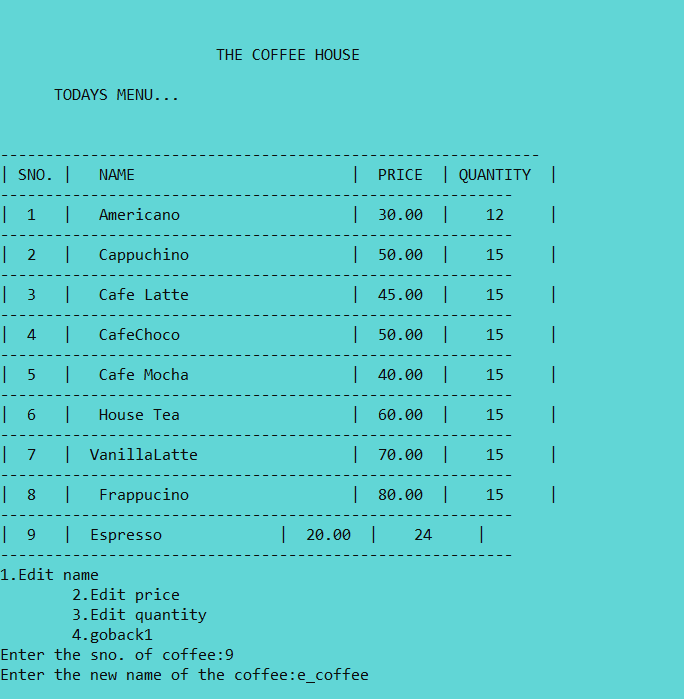




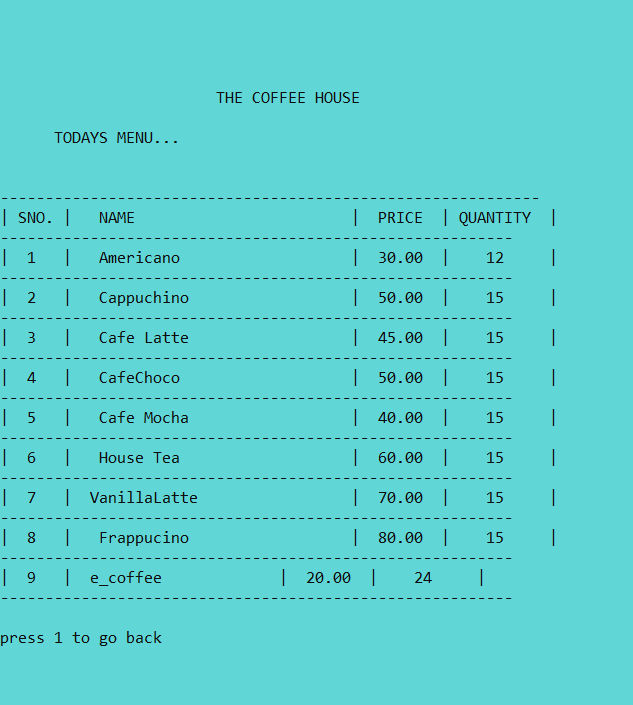
**If you enter 3**



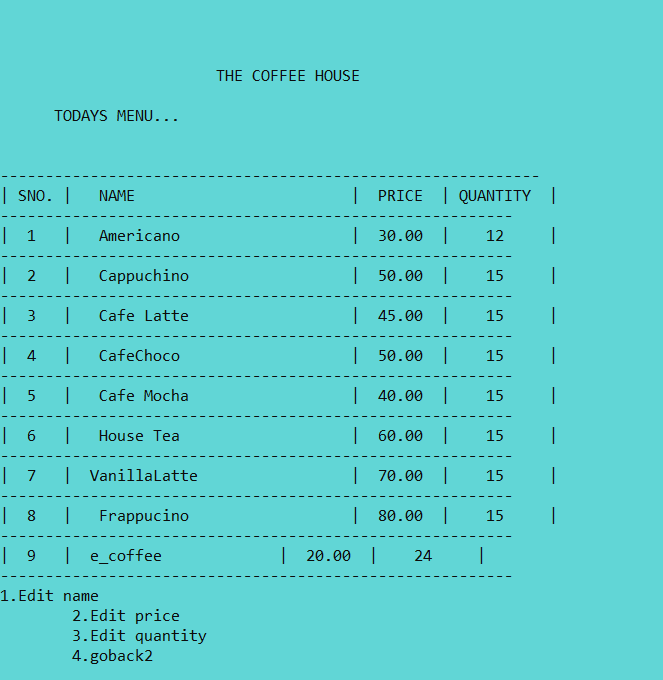
**Enter 1**



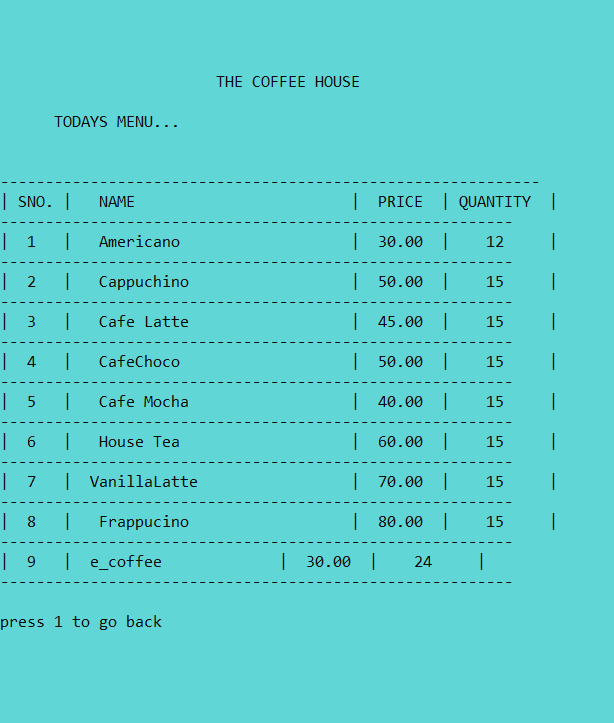
**Now enter display menu**



**Now select 2**

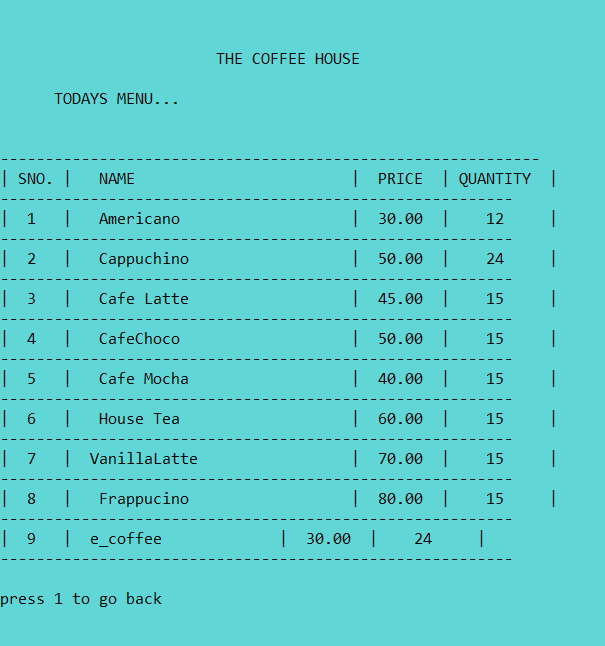




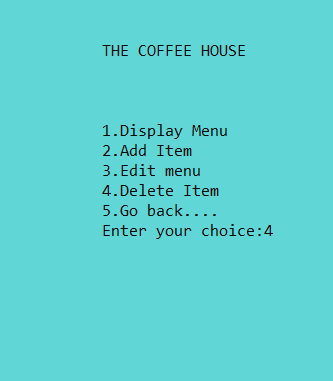


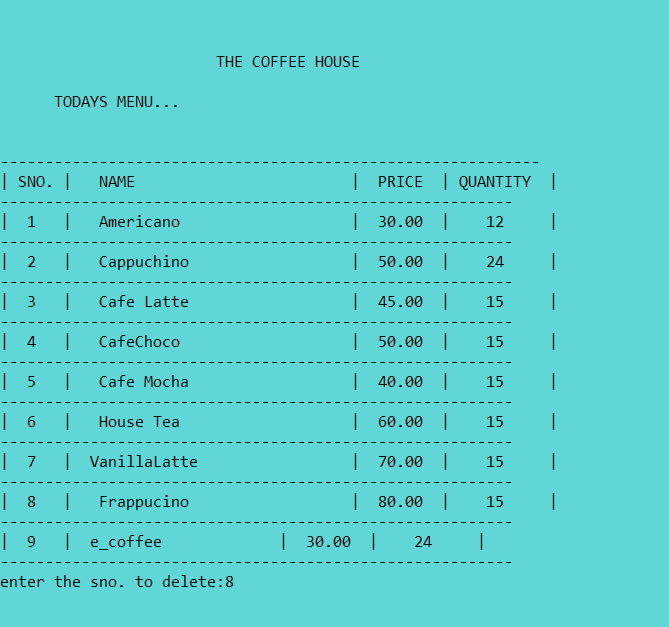






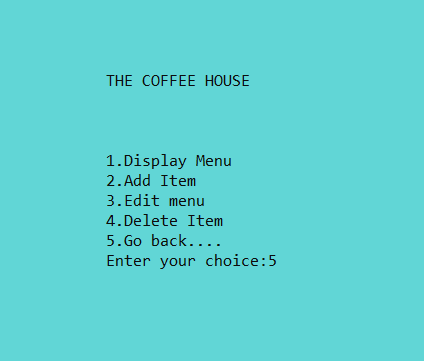
**Now enter 4**

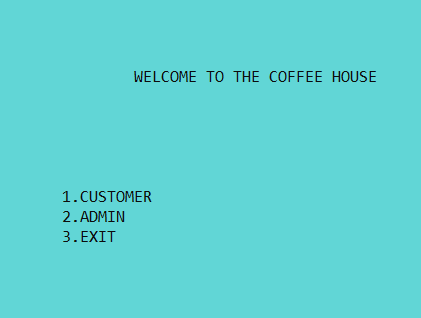




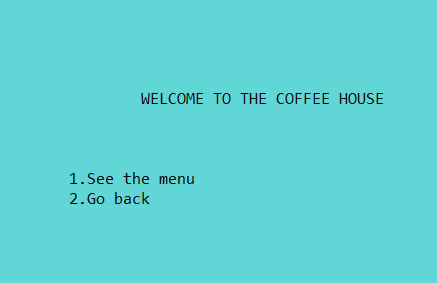
**Now the menu is**

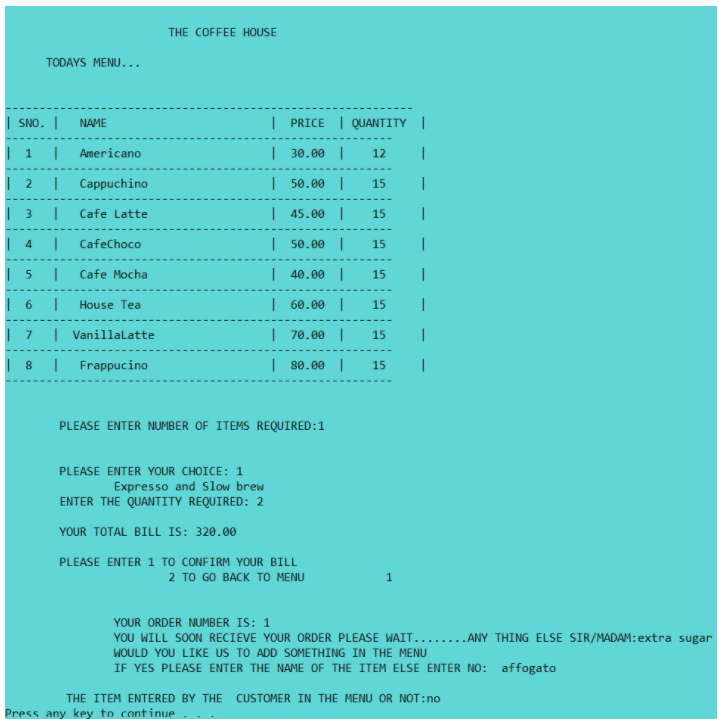


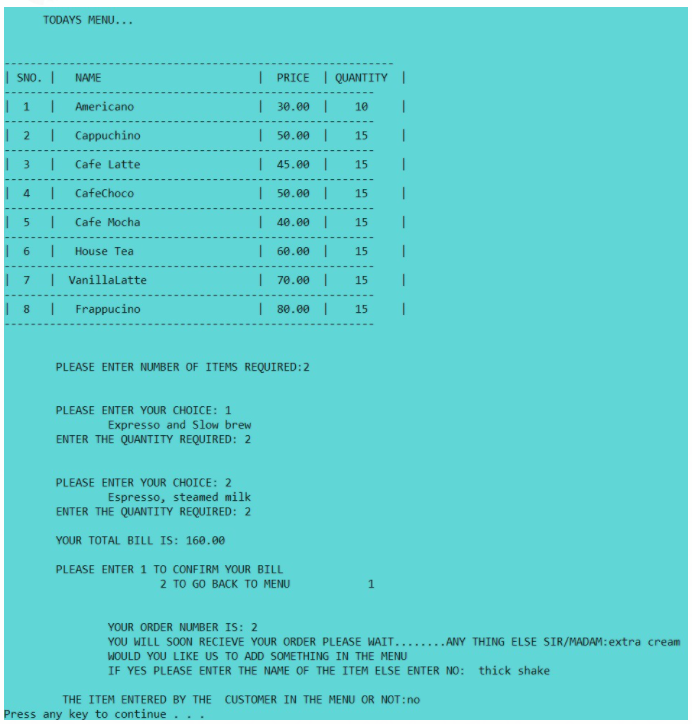


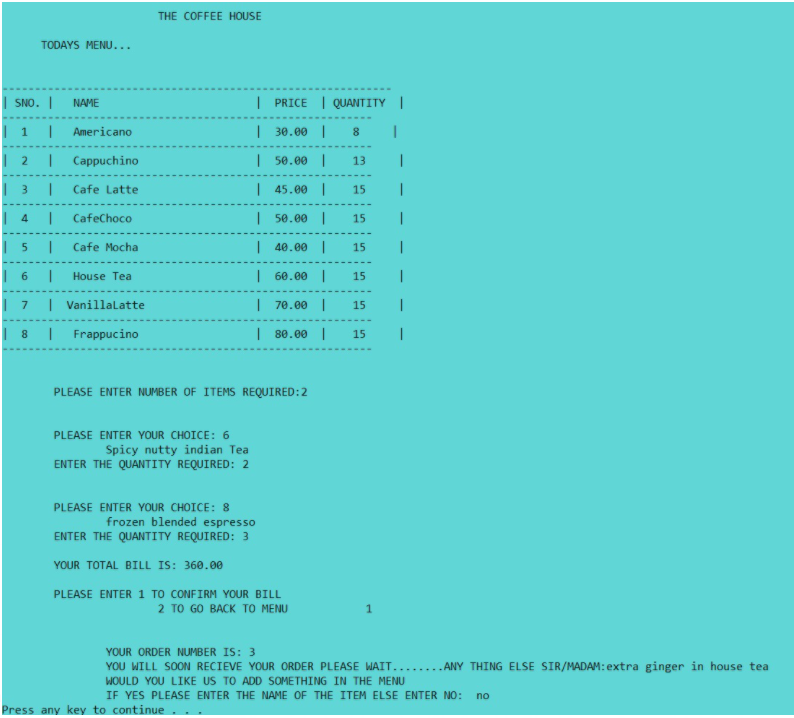


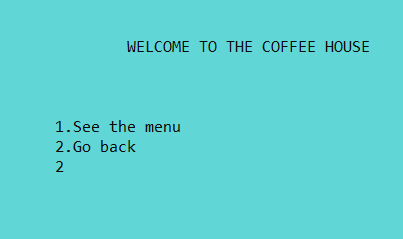
**Now enter 1**

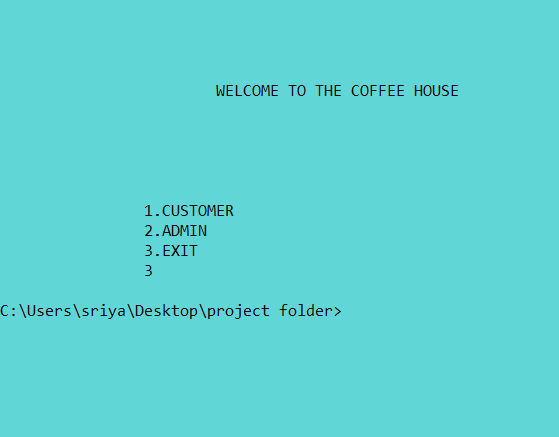




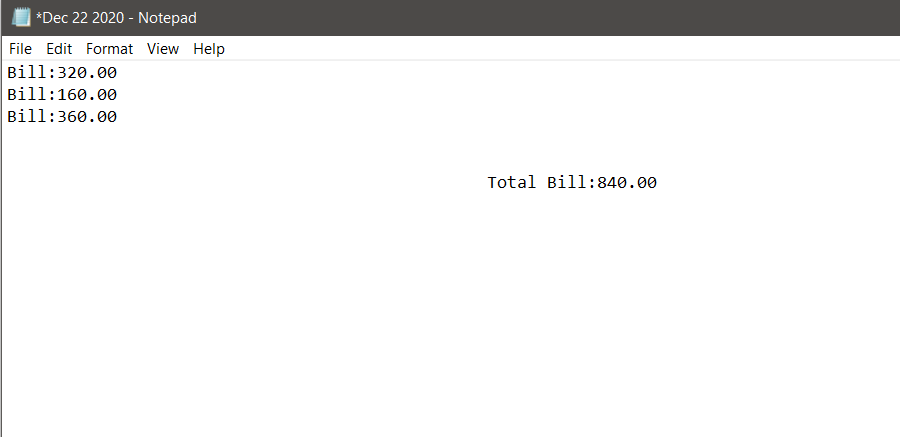


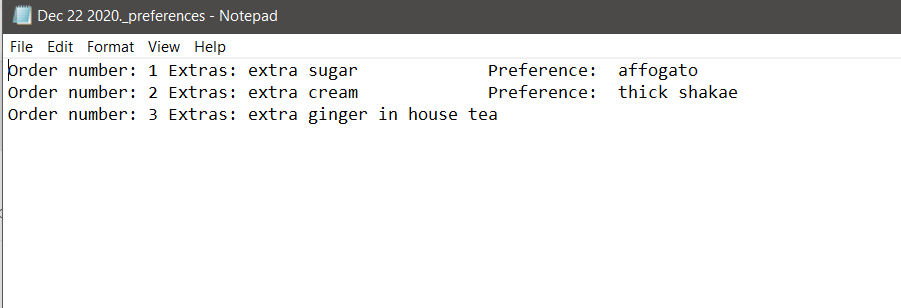






**Text files created on orders by customer with date:**





**5.0) ADDITIONAL KNOWLEDGE GAINED**

We gained practical knowledge by doing this project. It helped us to grasp the C concepts in depth. It gave us an opportunity to apply our knowledge on C in innovative ways. It strengthened the understanding of our fundamentals through effective application of theoretical concepts. Our project dealt with file handling and we have gained lot of improvement in this part. We have learned various concepts like time functions, how to organize different modules in a single program and also the structure concepts.We had a very good time learning and implementing graphics in our project.

**6.0) CONCLUSION**

We are living in a society where computer software has taken place over the paper and manual work. This project work is an attempt to develop a system that can be used for computerization of activities . Since these activities are tedious process requiring lot of effort, more care has been for the system development. Our project doesn’t require any paper work and created in user friendly environment. Data security and reliability is also maintained. It increases efficiency and saves time. This represents a typical real world situation. It helps the owner of coffee house to keep track of all the bills and staff to get to know the item preferred by customers. We have also added an additional feature of notifying the customer of the ingredients present in their orders. We have gained immense knowledge in coding and programming throughout this mini project. We have improved our collaborating skills and had fun while debugging and fixing errors. It was a great opportunity for both of us. The project was implemented and executed successfully.

**FUTURE ENHANCEMENT**

We can enhance this system by including more facilities like online booking, regular customer record. Further requirements and improvements can easily be done since the coding is mainly structured or modular in nature. Changing the existing modules or adding new modules can add improvements.

**7.0)** **REFERENCES**

We have referred to following sources for references during this project:

1. Computer Science: A structured Programming Approach Using C- Third Edition
2. [www.google.com](http://www.google.com)
3. [www.scribd.com](http://www.scribd.com)
4. [www.stackoverflow.com](http://www.stackoverflow.com)
5. [www.code-projects.org](http://www.code-projects.org)