

**A MINI PROJECT** **REPORT**

for

MINI PROJECT IN C(19CSE39)

**OptiO: FOOD DELIVERY SERVICE**

***submitted by***

**YASHAS KALATHUR ACHARYA**

**1NH19CS202**

**3/D**

***In partial fulfillment for the award of the degree of***

**BACHELOR OF ENGINEERING** **IN**

**COMPUTER SCIENCE AND ENGINEERING**



*Certificate*

This is to certify that the mini project work titled

OptiO:FOOD DELIVERY SERVCIE

Submitted in partial fulfillment of the degree of Bachelor of Engineering in Computer Science and Engineering

Submitted by

YASHAS KALATHUR ACHARYA

1NH19CS202

*DURING*

*ODD SEMESTER 2020-2021*

*FOR*

*19CSE39*

Signature of Reviewer Signature of HOD

SEMESTER END EXAMINATION

*Name of the Examiner Signature with date*

**1.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**2.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**ABSTRACT**

The Purpose of this project is to develop a food ordering service. It is a system that enables the customer to order different delicacies from various restraunts from the comfort of their homes. This system has not only helped the customers but has also helped the restraunt business. The number of orders have increased which has led to the increase of the profit. In the past the people had to go to the restraunts and either dine in or collect the food from the restraunt. But now with such a service, the food will arrive right at their doorstep. In this service, the user has two options. The first option is that the user can select items of their choice and also mention its quantity. The second option is that the user must specify how they are willing to spend and based on that the combos will be displayed and the user can select anyone of those combos. It gives you three different options of payments and the user can select from anyone of those methods. It also allows you to rate the service and give in your suggestions. The new feature in this project is the Option2 where the user has to input their budget and the two combos will be displayed.

**ACKNOWLEDGEMENT**

The satisfaction and euphoria that accompany the successful completion of any task would be impossible without the mention of the people who made it possible, whose constant guidance and encouragement crowned our efforts with success.

I have great pleasure in expressing gratitude to **Dr. Mohan Manghnani**, Chairman of New Horizon Educational Institutions for providing necessary infrastructure and creating good environment.

I take this opportunity to express my profound gratitude to **Dr. Manjunatha,** Principal NHCE, for his constant support and encouragement.

I would also like to thank Dr**. B. Rajalakshmi**, Professor and Head, Department of Computer Science and Engineering, for her constant support.

I express my gratitude to **Mr. Praveen Pawaskar,** Senior Assistant Professor**,** Department of Computer Science, my project guide, for constantly monitoring the development of the project and setting up precise deadlines. Her valuable suggestions were the motivating factors in completing the work.

**YASHAS KALATHUR ACHARYA**

**1NH19CS202**

**CONTENTS**

**ABSTRACT 3**

**ACKNOWLEDGEMENT 4**

**LIST OF FIGURES 6**

1. **INTRODUCTION** 
   1. PROBLEM DEFINITION **7**
   2. OBJECTIVES **7-8**
   3. METHODOLOGY TO BE FOLLOWED **8**
   4. EXPECTED OUTCOMES **8**
   5. HARDWARE AND SOFTWARE REQUIREMENTS **5**
2. **DATASTRUCTURES 6-7**
   1. LINKED LIST **7-9**
3. **DESIGN**
   1. ALGORITHM  **9-10**
   2. FLOWCHART **10-14**
4. **IMPLEMENATION** 
   1. MODULE1 **15-16**
   2. MODULE2 **16- 19**
   3. MODULE3 **19- 22**
   4. MODULE4 **22-28**
5. **RESULTS 28-33**
6. **CONCLUSION 34**
7. **REFERENCES 34**

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **Fig. No** | **Figure Description** | **Page No** |
| **1.1** | **Singly Linked List** | **11** |
| **1.2** | **Doubly Linked List** | **11** |
| **1.3** | **Circular Singly Linked List** | **11** |
| **1.4** | **Circular Doubly Linked List** | **12** |
| **2** | **Flowchart** | **14-17** |
| **3.1** | **Screenshot -1** | **31** |
| **3.2** | **Screenshot -2** | **32** |
| **3.3** | **Screenshot -3** | **32** |
| **3.4** | **Screenshot -4** | **33** |
| **3.5** | **Screenshot -5** | **34** |
| **3.6** | **Screenshot -6** | **34** |
| **3.7** | **Screenshot -7** | **35** |
| **3.8** | **Screenshot -8** | **35** |
| **3.9** | **Screenshot -9** | **36** |
| **3.10** | **Screenshot -10** | **36** |

**CHAPTER 1**

**INTRODUCTION**

The food ordering and delivery service is a courier service in which a user makes an order to a restaurant or any independent cook through his mobile phone or a website and gets it delivered to the place of your wish. The development of such a service has changed the whole restaurant business and has helped a lot of restaurants increase their income. Around 5 to 10 years ago, when we craved for restaurant food the only way to get it was go to that particular restaurant. But it wasn’t possible always as people had transport issues and other issues. The introduction of such a system has brought a revolution to the whole business.

**1.1 PROBLEM DEFINTION**

The food delivery service in the 21st century has revolutionised our access to various cuisines from different restaurants at any instant of the day. But did your brain ever get exhausted picking a combination of delicacies within your stipulated budget? OptiO is here to save your time by providing you options to select from so that you can bother about the important things in life

* 1. **OBJECTIVES**
* The general objective is to develop a reliable, convenient, and accurate service.
* The main objective to keep in mind is to provide excellent customer service. If the customer is not satisfied then the entire system will fail.
* To evaluate its performance and acceptability in terms of security, user friendliness, accuracy and reliability. While paying for the order, people pay using their debit or credit or through UPI. Such an instant where there is a risk of a fraud should not occur.
* The system will automatically calculate the total price and the user need waste their time on the calculations.
* To display the bill with all the details and giving a clear idea about the pricing of an item ordered by the user.
* To ensure easy and fast ordering process with no complex steps and should be understandable by everyone.
* To provide the delicious food for the user sitting at home and provide the service at any instant of the day.

**1.3 METHODOLOGY TO BE FOLLOWED**

The main objective of this project is to develop a mini project using Data Structure using C programming language. The Data Structure used in this particular mini project is Circular Singly Linked List. Circular Singly Linked List is a type of linked list where the first and last node are linked to each other. Here the linked list is used to store the data i.e., the item number, item name and the price of the menu of the three restaurants. Memory is allocated using Dynamic Memory Allocation. The function insertend() is used to create the node to store the data and function display() is used to display the menu of the restaurant selected by the user. The switch case statement is used to select the restaurant.

**1.4 EXPECTED OUTCOMES**

OptiO is an improved food delivery service. The user will be given two options. The first is the normal ordering process where the user has to specify the item number and the quantity he or she requires and based on that the final price will be printed. Along with the normal ordering process, there will be a feature where the user can specify how much he or she is willing to spend and based on that it will provide you with various combination of meals which include different items. For example, if the user is only willing to spend Rupees 500 then OptiO will give the user a selection of combination of food items from different food categories all under the specified amount. There will be a minimum amount that the user needs to specify in order to avail this feature. This will allow the user to save their time in the ordering process and make it much more user friendly.

**1.5 HARDWARE AND SOFTWARE REQUIREMENTS**

Concept Requirement

* Understanding the basics of C programming and Data Structures using C.

Software Requirement

1. Operating System: Windows 7 or above
2. Language Used: C programming Language
3. Front End/Back End: C compiler /Data Structures
4. Code Blocks

Hardware Requirements

1. Memory: 2GB RAM or more
2. Processor: Inter CORE processor
3. Hard Disk Space: 50 MB or more of memory space

**CHAPTER 2**

**DATA STRUCTURE**

* Data structure is the logical or mathematical model of a particular organization of data.i.e. how data is organised and stored in memory locations.
* Different kinds of data structures are suited to different kinds of applications and some are highly specialized to specific tasks.

**2.0.1 CLASSIFICATION OF DATA SRUCTURE**

The Classification of Data structure are:

1. **Primitive Data Structure**

* A primitive data structure is pre-defined by the programming language.
* They can be manipulated directly by machine instructions.
* They are also known as simple data structure.

1. **Non-Primitive Data Structure**

* They are derived from primitive data structure.
* Data structures that cannot be manipulated directly from the machine instructions.
* They are also known as Compound Data Structures.

**2.0.2 OPERATIONS ON DATA STRUCTURE**

* Create – Creating the initial data structure.
* Insert – Adding elements to the existing data structure.
* Delete – Remove and element from the data structure.
* Search – Find whether a particular element is present in the data structure or not.
* Sort – Arranging the elements of the data structure in ascending or descending order.
* Traverse – Accessing each element of the data structure one by one.
  + 1. **TYPES OF DATA STRUCTURE**
* Arrays
* Linked Lists
* Stacks
* Queues
* Trees
* Graphs

**2.1 LINKED LIST**

Linked list is a dynamic data structure which consists of non – sequential collection of data types. We call linked lists as linear data structure because of the appearance of it i.e.linked list is collection of items where each item contains data and pointer or link of next item.

**2.1.1 ADVANTAGES OF LINKED LISTS**

* Linked Lists are dynamic i.e. they can grow or shrink during execution of the program
* Memory is not pre-allocated in linked lists hence it encourages efficient usage of memory.
* Insertion and deletion can be easily carried over.
* Many complex applications can be done using linked lists.

**2.1.2 DISADVANTAGES OF LINKED LISTS**

* Consumes more memory because every node requires additional pointer to store the address of the next node.
* Searching a particular item is time consuming.

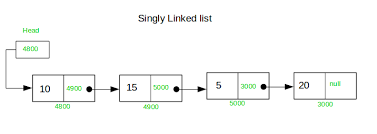
**2.1.3 TYPES OF LINKED LIST**

**i) SINGLE LINKED LIST**

* A single linked list is a collection of nodes where each node has two parts:

1. Data
2. Link- Holds address of the next node

* The first nodes address is stored in a pointer variable called Head and the last nodes pointer has NULL.

Fig 1.1

**ii)DOUBLE LINKED LIST**

* It is a collection of nodes where each node has three parts, pointer to the previous element, data and the pointer to the next element in the list.
* Through Double Linked List, we can traverse the list of the elements in both the directions.

Fig 1.2

**iii) CIRCULAR LINKED LIST**

* In any linked list, if the last node is pointed back to the first node then it is called Circular Linked List. It can be divided into two types

1. **Circular Singly Linked List**

* The last node of the list contains a pointer to the first node .There is no NULL value present in the next part of the nodes

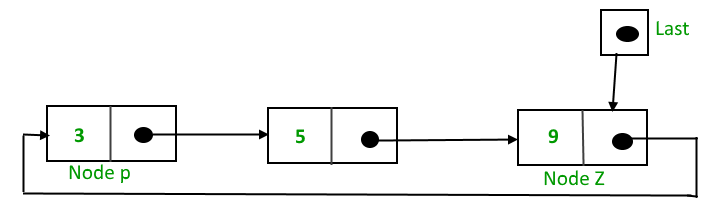
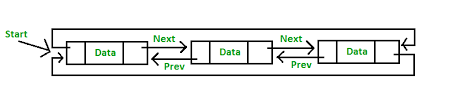


Fig 1.3

1. **Circular Doubly Linked List**

* The last node of the list contains a pointer to the first node. There is no NULL value present in the next part of the nodes

 Fig 1.4

**CHAPTER 3**

**DESIGN**

**3.1 ALGORITHM**

An algorithm is a step-by-step process that defines a set of instructions to be executed in a specific order in order to achieve the required result. An algorithm can be expressed within a finite amount of space and time and in a well-defined formal language for calculating a function.

**3.2.1 ADVANTAGES OF ALGORITHM**

* It is a step-wise representation of a solution to a given problem, which makes it easy to understand.
* An algorithm uses a definite procedure.
* It is not dependent on any programming language, so it is easy to understand for anyone even without programming language.
* Every step in an algorithm has its own logical sequence so it is easy to debug.
* By using algorithm, the problem is broken down into smaller pieces, hence it is easier for the programmer to convert it into an actual program.

**3.2.2 DISADVANTAGES OF ALGORITHM**

* Algorithms is time consuming.
* Difficult to show Branching and Looping in Algorithms.
* Big tasks are difficult to put in Algorithms.

**3.2.3 STEPS OF THE ALGORITHM**

1. **Start**
2. **Enter user name, phone number and address.**
3. **Select restaurant from the given list of restaurants.**
4. **If user selects first restaurant, menu of that restaurant will be displayed.**
5. **If user selects second restaurant, menu of second restaurant will be displayed.**
6. **If user selects third restaurant, menu of third restaurant will be displayed.**
7. **User should select one of the options: either option 1 or option 2.**
8. **If user selects option 1:**
9. **User should select item number and item price.**
10. **User will be asked if they want to continue ordering.**
11. **If yes, step A and B will repeat.**
12. **If no, then the bill will be displayed.**
13. **If the user selects option 2**
14. **The user will be asked how much they are willing to pay.**
15. **The user will input the price.**
16. **Based on the price mentioned by the user, various combination of meals will be displayed.**
17. **User selects anyone of the given combinations.**
18. **The bill will be displayed.**
19. **The mode of payment will be asked.**
20. **If the user wants to pay through Debit or Credit card.**
21. **Card Number and CVV will be asked.**
22. **OTP to be entered.**
23. **Or User can pay through UPI. Phone number will be asked from the user.**
24. **Registered Phone number will be asked.**
25. **OTP to be entered.**
26. **Else cash on delivery**
27. **Once ordering process is done, the time required for the order to arrive at the specified place will be displayed.**
28. **The user will then be asked if the order has arrived or not.**
29. **If the order hasn’t arrived then the user will be told that the order is on its way.**
30. **Once the order arrives, the user will be asked to give a rating out of 5 to the service and also give a feedback.**
31. **Message of gratitude will also be displayed.**
32. **Exit**

**3.2 FLOWCHART**

Flowchart is a graphical representation of an algorithm. Programmers often use it as a program planning tool to solve a problem. It makes use of symbols which are connected among them to indicate the flow of information and processing.

* + 1. **ADVANTAGES OF FLOWCHART**
* Easier to understand compared to algorithms and pseudo code.\
* Helps us to understand Logic of the problem.
* It is very easy to draw flowchart.
* Using only very few symbols, complex problems can be solved.
  + 1. **DISADVANTAGES OF FLOWCHART**
* Making alterations and modifications is hard.
* Reproduction of flowchart becomes a problem.
* Flowchart can become complex and clumsy depending on the logic.

Enter name, phone number and address

User should select restaurant

Switch

(Choice)

Display menu of 1st restaurant

Choice1

Display Menu of 2nd restaurant

Choice 2e 2Restaurant 2

Choice 3

Display Menu of 3rd restaurant

Select option 1 or 2

If User selects option 1

**FALSE**

**TRUE**

User enters stipulated amount

User selects item number and quantity quantity

Combos will be displayed

Ask user to Continue to order or not

User selects the combo

If yes

**TRUE**

**FALSE**

Display Bill

Select Mode of payment

switch(payment)

Debit card

Enter card number, CVV and OTP

Enter phone number and OTP

UPI

Cash on delivery

Required time for arrival of order.

Ask if order received

or not

if yes

**FALSE**

**TRUE**

Ask the user to give a rating out of 5

Message of gratitude will be printed

Ask user to give his feedback regarding the service

Fig 2

**CHAPTER 4**

**IMPLEMENTATION**

**4.1) MODULE 1**

The main function includes the function call to display the menu of a particular restraunt, the mode of payment through which the user wants to make the payment and the rating of the service.

void main()

{

char name[30],add[100],suggests[100];

char phno[10],cardno[30],cvv[5],phno1[10];

int confirm,rate;

int x,op,inum,q,mode,otp,diff;

float y;

char fooditem[30];

int choice;

printf("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ WELCOME TO OPTIO ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~");

printf("\nEnter your name: ");

getchar();

gets(name);

printf("\nEnter your phone number: ");

while(1)

{

scanf("%s",phno);

if(strlen(phno)!=10)

{

printf("Error!Re Enter phone number: ");

continue;

}

break;

}

printf("\nEnter your address: ");

getchar();

gets(add);

printf("\n~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ RESTAURANTS ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~\n");

printf("\n1.The Burger Place \n2.Indian Delight \n3.The Pizzeria\n");

printf("\nSELECT A RESTRAUNT: ");

while(1)

{

scanf("%d",&choice);

if(choice!=1 && choice!=2 && choice!=3)

{

printf("\nRe Enter your choice: ");

continue;

}

break;

}

switch(choice)

{

case 1: restraunt1();

break;

case 2: restraunt2();

break;

case 3 :restaurant3();

break;

default:printf("INVALID OPTION!\n");

}

* The structure will include item number, foodname, quantity, item price and the link which is used to create the menus which is done using Cicular Singly Linked List.

struct node

{

int itemno,quantity;

char foodname[30];

float price;

struct node \*link;

};

**4.2 MODULE 2**

* The insert function is used to create the nodes of the linked list which will store the item number, foodname, quantity, item price of the restaurants and will also include the address of the next nodes.

NODE insert(NODE head,int itemno,char foodname[30],float price)

{

NODE temp,cur;

temp=(NODE\*)malloc(sizeof(struct node));

temp->itemno=itemno;

temp->price=price;

strcpy(temp->foodname,foodname);

temp->link=temp;

cur=head;

if(cur->link==NULL)

{

head->link=temp;

temp->link=head;

}

else

{

while(cur->link!=head)

{

cur=cur->link;

}

cur->link=temp;

temp->link=head;

}

return head;

}

* The restraunt1, restraunt2 and restraunt3 functions will call the insert function and the elements will be stored in the nodes of the linked list. It also includes the display function call,OPTION1 and OPTION2 function calls.

void restraunt1()

{

int op,inum,q,combo;

int choice;

float amount,totalamount=0;

head=NULL;

head=insert(&head,1,"Small Fries",100);

head=insert(&head,2,"Medium Fries",150);

head=insert(&head,3,"Large Fries",175);

head=insert(&head,4,"Veg Burger-S",250);

head=insert(&head,5,"Veg Burger-L",300);

head=insert(&head,6,"Chikn Burger-S",350);

head=insert(&head,7,"Chickn Burger-L",400);

head=insert(&head,8,"Small Pepsi",100);

head=insert(&head,9,"Medium Pepsi",125);

head=insert(&head,10,"Large Pepsi",150);

display(head);

printf("\nOption 1: Select any item from the menu of your choice\n");

printf("\nOption 2: Enter your budget and select any combo from the two combos that will be displayed\n");

printf("\nSelect Option 1 or Option 2: ");

while(1)

{

scanf("%d",&op);

if(op!=1 && op!=2)

{

printf("\nRe Enter your choice: ");

continue;

}

break;

}

if(op==1)

OPTION1();

if(op==2)

OPTION2();

}

void restraunt2()

{

int op,choice,inum,q,combo;

float amount,i1,i2,i3;

float totalamount=0;

head=NULL;

head=insert(&head,1,"Paneer 65",100);

head=insert(&head,2,"Chicken 65",150);

head=insert(&head,3,"Chikn Tikka(4)",175);

head=insert(&head,4,"Veg Biryani",250);

head=insert(&head,5,"Spl Veg Biryani",300);

head=insert(&head,6,"Chikn Biryani",350);

head=insert(&head,7,"Sp Chkn Biryani",400);

head=insert(&head,8,"Vanilla Shake",100);

head=insert(&head,9,"Chocolate Shake",125);

head=insert(&head,10,"Oreo Shake",150);

display(head);

printf("\nOption 1: Select any item from the menu of your choice\n");

printf("\nOption 2: Enter your budget and select any combo from the two combos that will be displayed\n");

printf("\nSelect Option 1 or Option 2: ");

while(1)

{

scanf("%d",&op);

if(op!=1 && op!=2)

{

printf("\nRe Enter your choice: ");

continue;

}

break;

}

if(op==1)

OPTION1();

if(op==2)

OPTION2();

}

void restaurant3()

{

int op,choice,inum,q,combo;

float amount;

float totalamount=0;

head=NULL;

head=insert(&head,1,"French Fries",100);

head=insert(&head,2,"Potato Fingers",150);

head=insert(&head,3,"Garlic Bread(4)",175);

head=insert(&head,4,"Veg Exotica",250);

head=insert(&head,5,"Very Veggie",300);

head=insert(&head,6,"Chikn Pizza",350);

head=insert(&head,7,"Pepperoni Pizza",400);

head=insert(&head,8,"Small Pepsi",100);

head=insert(&head,9,"Medium Pepsi",125);

head=insert(&head,10,"Large Pepsi",150);

display(head);

printf("\nOption 1: Select any item from the menu of your choice\n");

printf("\nOption 2: Enter your budget and select any combo from the two combos that will be displayed\n");

printf("\nSelect Option 1 or Option 2: ");

while(1)

{

scanf("%d",&op);

if(op!=1 && op!=2)

{

printf("\nRe Enter your choice: ");

continue;

}

break;

}

if(op==1)

OPTION1();

if(op==2)

OPTION2();

}

* Once the display function is called in one of the three restraunt functions, the display function will access the elements from the linked list and will display it in the form of a menu of a particular restraunt.

void display(NODE head)

{

NODE temp;

temp=head->link;

printf("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ THE MENU ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~\n");

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

printf("\t\t\t\t| ITEM NO.\t|\t NAME\t\t| TOTALPRICE |\n");

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

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

while(temp!=head)

{

printf("\t\t\t\t|\t%d\t|\t%s\t|\t%0.2f\t|\n",temp->itemno,temp->foodname,temp->price);

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

temp=temp->link;

}

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

}

**4.3 MODULE 3**

* The user has two choices when it comes to ordering process. The first option is to select the item of your choice and its quantity. The user will enter the respective and it will be then passed to the calculation1 function where the final price will be calculated and returned to the function.

void OPTION1()

{

int op,choice,inum,q,combo;

float amount;

float totalamount=0;

while(1)

{

printf("\nEnter the item number: ");

scanf("%d",&inum);

printf("\nEnter the quantity: ");

scanf("%d",&q);

amount=calculation1(&head,inum,q);

totalamount+=amount;

printf("\nCurrent total is:%0.2f\n",totalamount);

printf("\n1)Continue ordering 0)Exit\n");

printf("\nEnter your choice: ");

scanf("%d",&choice);

if(choice==1)

continue;

else

{

printf("\n~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ BILL ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~\n");

displayorder(&head1,q);

printf("\nTotal Amount:%0.2f\n",totalamount);

printf("~~~~~~~~~~~~~~~~~~~~~~");

break;

}

}

}

* The calculation1 function will traverse the linked list and access the item that was mentioned by the user. The calculations of the price will also be done in this function and the item number, price, foodname and quantity will be passedto the order function where the respective items are stored in the linked list.

float calculation1(NODE head,int inum,int q)

{

NODE cur;

int itemno,quantity;

cur=head->link;

float x;

float price;

char foodname[30];

while(cur!=head)

{

if(inum==cur->itemno)

{

x=cur->price;

x=x\*q;

itemno=cur->itemno;

price=cur->price;

strcpy(foodname,cur->foodname);

head1=order(&head1,itemno,foodname,x,q);

return x;

}

cur=cur->link;

}

}

* The order function will create another linked list where the items from the calculation1 function will be passed and stored in the linked list. It will then return to the calculation1 function and then the calculation1 function will return the price value to the OPTION1 function.

NODE order(NODE head1,int itemno,char foodname[30],float price,int q)

{

NODE temp,cur;

temp=(NODE\*)malloc(sizeof(struct node));

temp->itemno=itemno;

temp->price=price;

strcpy(temp->foodname,foodname);

temp->quantity=q;

temp->link=NULL;

cur=head1;

if(cur->link==NULL)

{

head1->link=temp;

temp->link=head1;

}

else

{

while(cur->link!=head1)

{

cur=cur->link;

}

cur->link=temp;

temp->link=head1;

}

return head1;

}

* The OPTION1 function will call the displayorder function and all the elements will be passed to the respective function and the items that were selected by the user will be displayed along with the final price wil be displayed which will be the bill.

void displayorder(NODE head1,int q)

{

NODE temp;

temp=head1->link;

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

printf("\t\t|ITEM NO.\t|\tNAME\t\t|\tQUANTITY\t|\tTOTALPRICE|\n");

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

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

while(temp!=head1)

{

printf("\t\t|\t%d\t|\t%s\t|\t%d\t\t|\t%0.2f\t |\n",temp->itemno,temp->foodname,temp->quantity,temp->price);

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

temp=temp->link;

}

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

}

**4.4 MODULE 4**

* If the user doesn’t select option 1, the user will select option 2. In the OPTION2 function the user will be asked their budget and based on that price the two different combos will be displayed. Once the user enters the budget, the formula1 and formula2 function will be called where the budget will be passed.

void OPTION2()

{

float budget;

int combo;

printf("\nPlease mention your budget in the range of 500 to 800 Rupees\n");

printf("\nEnter your budget: ");

while(1)

{

scanf("%f",&budget);

if(budget<500 || budget>800)

{

printf("\nRe Enter your budget: ");

continue;

}

break;

}

formula1(budget);

formula2(budget);

printf("\n\nWhich Combo would you like to order?\n");

printf("\n1)COMBO 1\n2)COMBO 2\n");

printf("\nSelect the COMBO: ");

while(1)

{

scanf("%d",&combo);

if(combo==1)

{

formula1(budget);

break;

}

if(combo==2)

{

formula2(budget);

break;

}

else

{

printf("\nError!Re enter your choice: ");

continue;

}

}

}

* In the formula1 and formula2 functions, the price for different items will be calculated and theese prices will be passed to the combo1 and combo2 functions.

void formula1(float budget)

{

float i1,i2,i3;

i1=budget \* (0.2);

i2=budget \* (0.5);

i3=budget \* (0.1);

combo1(budget,i1,i2,i3);

}

void formula2(float budget)

{

float i1,i2,i3;

i1=budget \* (0.2);

i2=budget \* (0.4);

i3=budget \* (0.2);

combo2(budget,i1,i2,i3);

}

* These two functions will traverse the linked list and will display the best items that staisfy the conditions. The two combos will then be displayed. It will then return to the OPTION2 function where the user has to make a choice between the two combos. Whichever combo is selected by the user will be displayed in the screen.

void combo1(float budget,float i1,float i2,float i3)

{

NODE cur,item1,item2,item3;

cur=head->link;

float diff,price,tprice,d;

while(cur->itemno<=3)

{

price=cur->price;

diff=price-i1;

if(fabs(diff)<25)

{

item1=cur;

}

cur=cur->link;

}

while(cur->itemno<=7)

{

price=cur->price;

diff=price-i2;

if(fabs(diff)<50)

{

item2=cur;

}

cur=cur->link;

}

while(cur->itemno<=10)

{

diff=cur->price-i3;

if(fabs(diff)<=70)

{

item3=cur;

}

cur=cur->link;

}

tprice=item1->price+item2->price+item3->price;

printf("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ COMBO 1 ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~\n\n");

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

printf("\t\t\t\t|ITEM NO.\t|\tNAME\t\t|TOTALPRICE\t|\n");

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

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

printf("\t\t\t\t|\t%d\t|\t%s\t|\t%0.2f\t|\n",item1->itemno,item1->foodname,item1->price);

printf("\t\t\t\t|\t%d\t|\t%s\t|\t%0.2f\t|\n",item2->itemno,item2->foodname,item2->price);

printf("\t\t\t\t|\t%d\t|\t%s\t|\t%0.2f\t|\n",item3->itemno,item3->foodname,item3->price);

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

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

printf("Total price:%0.2f\n",tprice);

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

d=budget-tprice;

printf("\nAmount saved:%0.2f\n",d);

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

}

void combo2(float budget,float i1,float i2,float i3)

{

NODE cur,item1,item2,item3;

cur=head->link;

float diff,price,tprice,d;

while(cur->itemno<=3)

{

price=cur->price;

diff=price-i1;

if(fabs(diff)<50)

{

item1=cur;

}

cur=cur->link;

}

while(cur->itemno<=7)

{

price=cur->price;

diff=price-i2;

if(fabs(diff)<51)

{

item2=cur;

}

cur=cur->link;

}

while(cur->itemno<=10)

{

diff=cur->price-i3;

if(fabs(diff)<=70)

{

item3=cur;

}

cur=cur->link;

}

tprice=item1->price+item2->price+item3->price;

printf("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ COMBO 2 ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~\n\n");

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

printf("\t\t\t\t|ITEM NO.\t|\tNAME\t\t|TOTALPRICE\t|\n");

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

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

printf("\t\t\t\t|\t%d\t|\t%s\t|\t%0.2f\t|\n",item1->itemno,item1->foodname,item1->price);

printf("\t\t\t\t|\t%d\t|\t%s\t|\t%0.2f\t|\n",item2->itemno,item2->foodname,item2->price);

printf("\t\t\t\t|\t%d\t|\t%s\t|\t%0.2f\t|\n",item3->itemno,item3->foodname,item3->price);

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

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

printf("Total price:%0.2f\n",tprice);

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

d=budget-tprice;

printf("\nAmount saved:%0.2f\n",d);

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

}

* It will then return to the main function where the user will asked to select the mode of payment and will be asked to give a rating to the service.

printf("\n\nEnter the mode of payment\n");

printf("\n1)Cash on Delivery\n2)Debit Card\n3)UPI\n");

printf("\nMode of payment: ");

while(1)

{

scanf("%d",&mode);

if(mode!=1 && mode!=2 && mode!=3)

{

printf("Re Enter the correct choice: ");

continue;

}

break;

}

switch(mode)

{

case 1: break;

case 2: printf("\nEnter the Debit Card number: ");

while(1)

{

scanf("%s",cardno);

if(strlen(cardno)!=16)

{

printf("\nError!Re enter your Debit Card no: ");

continue;

}

break;

}

printf("\nEnter the CVV number: ");

while(1)

{

scanf("%s",cvv);

if(strlen(cvv)!=3)

{

printf("\nError!Re enter your CVV: ");

continue;

}

break;

}

printf("\nEnter the generated OTP: ");

while(1)

{

scanf("%s",otp);

if(strlen(otp)!=4)

{

printf("\nError!Re Enter your OTP: ");

continue;

}

break;

}

printf("\nYour payment has been processed!!");

break;

case 3: printf("\nEnter the phone no: ");

while(1)

{

scanf("%s",phno1);

diff=strcmp(phno,phno);

if(diff!=0)

{

printf("Error!Please enter the right phone number: ");

continue;

}

break;

}

printf("\nEnter the OTP generated: ");

while(1)

{

scanf("%s",otp);

if(strlen(otp)!=4)

{

printf("\nError!Re enter your OTP: ");

continue;

}

break;

}

printf("\nYour payment has been processed!!");

break;

default: printf("Inavlid Choice!!!");

break;

}

printf("\nYour order will be delivered to your address in the next 30 minutes\n");

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

printf("\nHave you received your order?\n");

printf("\n1)Yes\n2)No\n");

printf("\nEnter the choice: ");

while(1)

{

scanf("%d",&confirm);

if(confirm==2)

{

printf("Please wait your order will arrive soon!\n");

printf("\nHave you received your order?\n");

printf("\n1)Yes\n2)No\n");

printf("\nEnter the choice: ");

continue;

}

break;

}

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

printf("\nPlease rate the service out of 5!\n");

printf("\n1-Unsatisfactory\n2-Satisfactory\n3-Good\n4-Very Good\n5-Excellent\n");

printf("\nRATING: ");

while(1)

{

scanf("%d",&rate);

if(rate>5)

{

printf("Enter the correct choice; ");

continue;

}

break;

}

printf("\n\nPlease suggest in what aspects we can improve\n");

getchar();

gets(suggests);

printf("\nThank you for the feedback.All points will be noted down and will be taken into consideration!!\n");

printf("\n~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~Thank You for ordering from Optio~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~\n");

**CHAPTER 5**

**RESULTS**

* The first three images will dsiplay the output where the user has to enter his details and once they select anyone of the restaurants, the menu of the selected restraunt will be displayed.



Fig 3.1

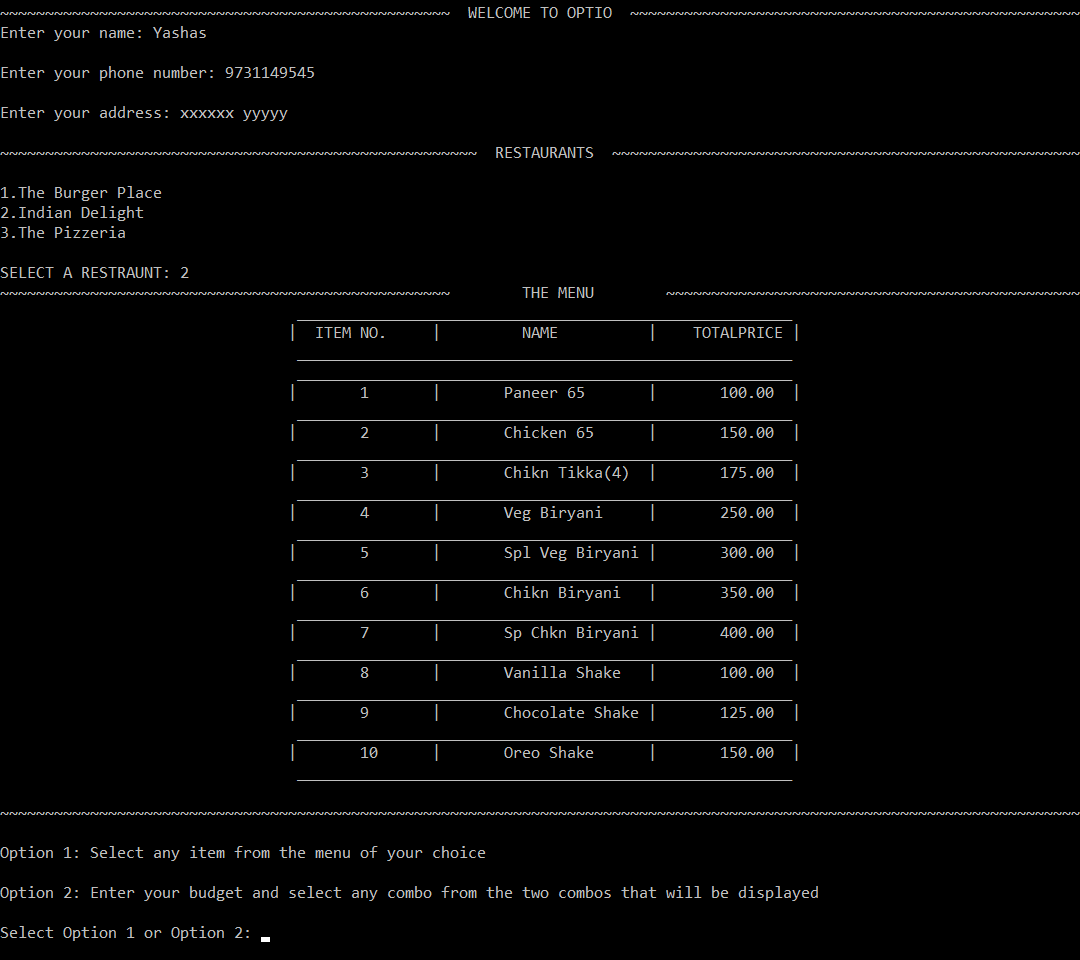


Fig 3.2



Fig 3.3

* The fourth image consists of the output when the user selects Option1. The user will be asked to enter the item number and quantity that they want to order and when the user is done, it will display the final bill consisting of the items ordered and the final price.

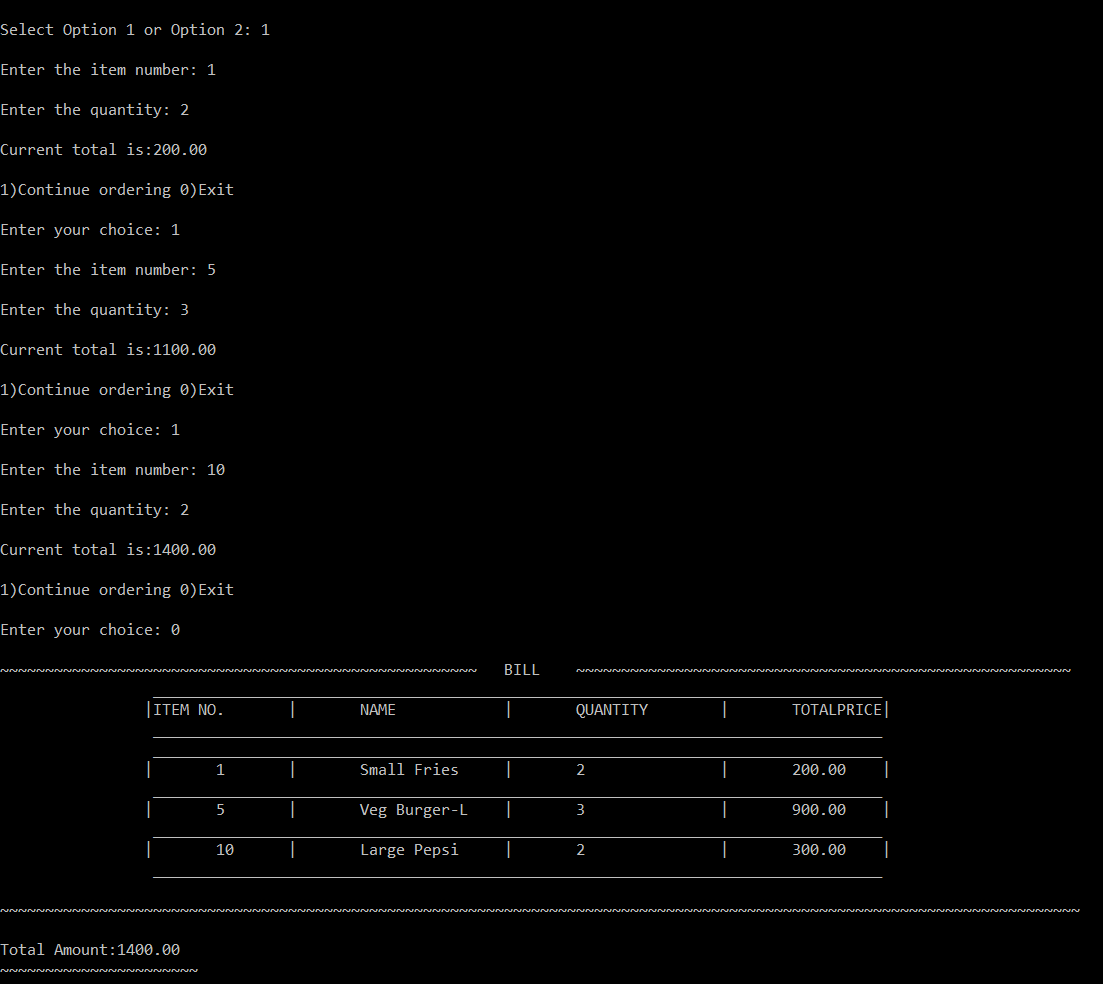


Fig 3.4

* The fifth, sixth and seventh images consists of the output when the user selects Option2. The user is supposed to enter their budget and based on tat budget two combos will be displayed and the user must select anyone of those two combos. The combo which is selected by the user will be displayed along with the final price and amont saved.

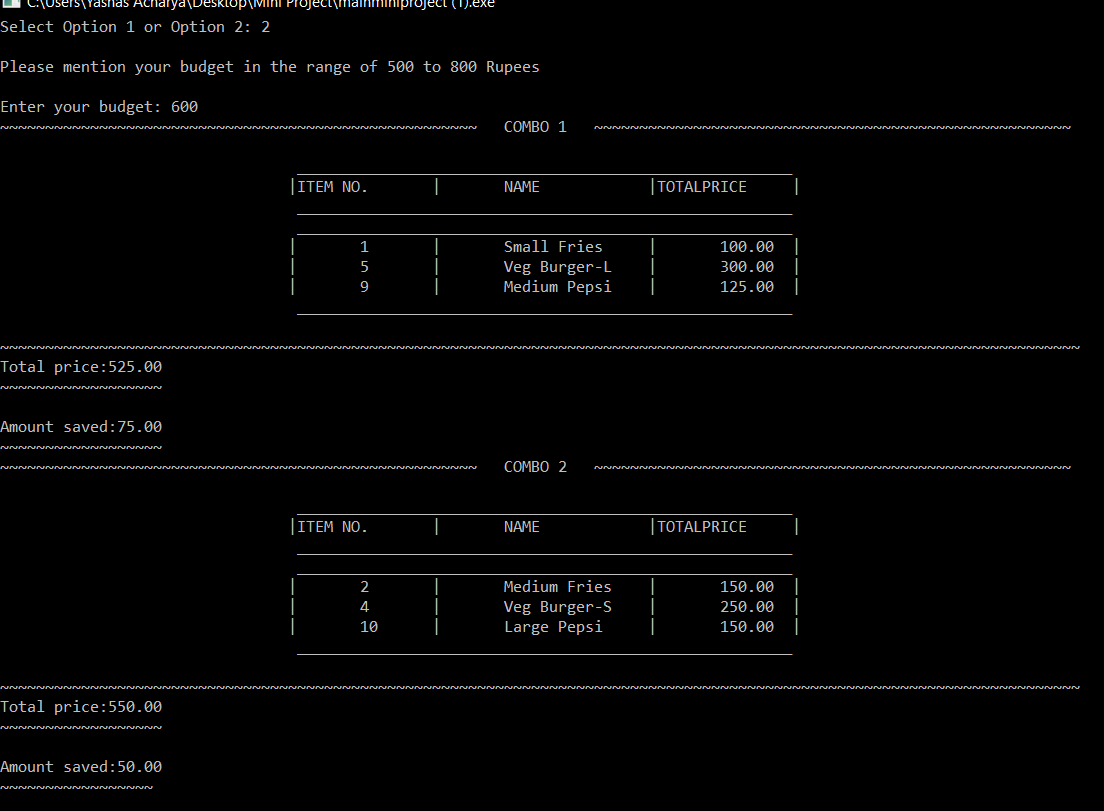


Fig 3.5

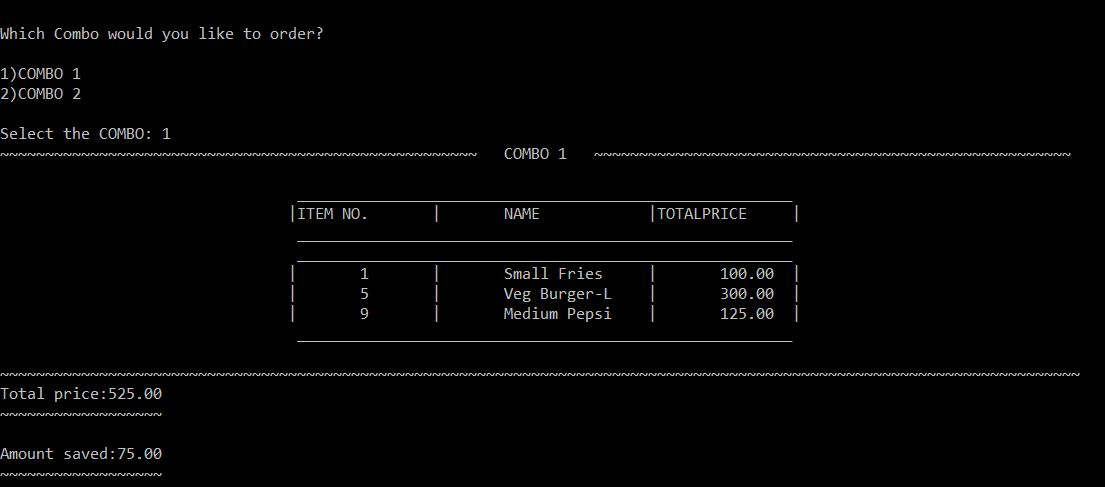


Fig 3.6

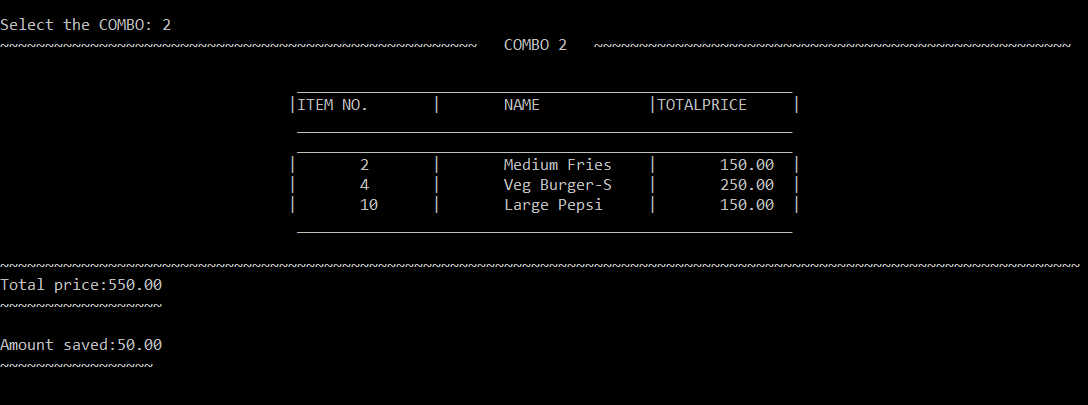


Fig 3.7

* The eighth, ninth and tenth image will display the various modes of payments that the user can select from and the rating service in order to rate and give certain feedbacks to the service.



Fig 3.8

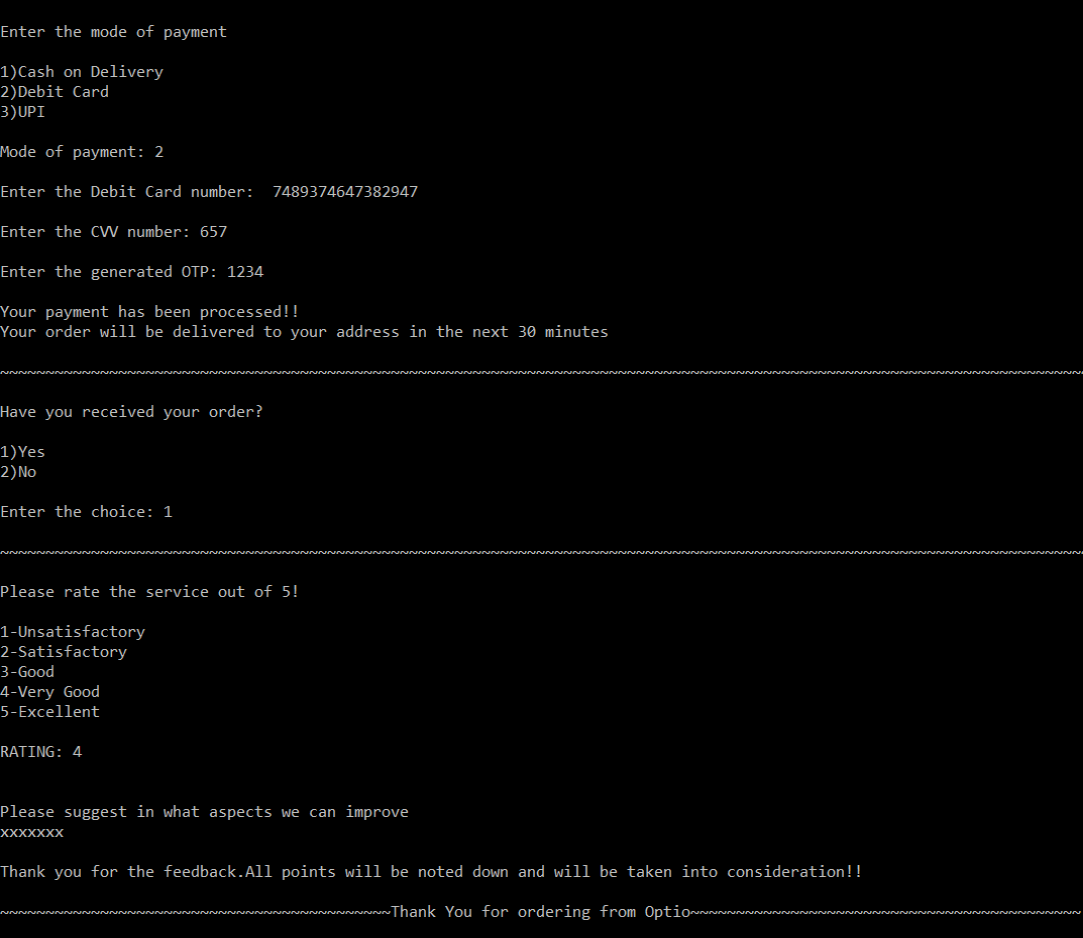
Fig 3.9



Fig 3.10

**CHAPTER 6**

**CONCLUSION**

From the above we can conclude that the following service is the future of the restraunt business. It not only helps the big restaurants but it will also help the small restaurants and the independent cooks to expand their business. This will also increase the employnment in the country as the demand for delivery people will also increase. The availability of restraunt food at any time and at any place is what most people want. If we are busy and we may not have the time to cook, we can just open the service and order and it will arrive to your doorstep. This service is not only beneficial for the customers but it also is helpful for the restraunt business. I belive we should encourage such services and find ways to make it much better and efficient in order t expand it and increase the revenue.

**CHAPTER 7**

**REFERENCES**

1. <https://youtu.be/7eWzvDlOVJ4>
2. <https://github.com/IstiyakV/Food-Order-Management-System>
3. https://www.geeksforgeeks.org/