A Mini Project Report

on

**FIND MY SHOP**

**Course:** Data Structures Lab

Sem: III Sec: CSE-B

By

NK.Pranesh

1602-19-733-088

L.Prabhu Joth

1602-19-733-087



**Department of Computer Science & Engineering**

**Vasavi College of Engineering (Autonomous)**

**Ibrahimbagh, Hyderabad-31**

**2020**

**Acknowledgment**

We respect and thank Dr. V.Sireesha mam for constant support and motivation which helped us immensely in completing our project successfully.

|  |  |  |
| --- | --- | --- |
| 1 | Abstract | 4 |
| 2 | Introduction | 5 |
| 3 | Design | 6 |
| 4 | Code/Implementation | 7-22 |
| 5 | Outputs | 23-24 |
| 6 | Test cases | 24-25 |
| 7 | Conclusion | 25 |
| 8 | References | 25 |

**TABLE OF CONTENTS:**

|  |  |  |
| --- | --- | --- |
| **S.No** | **Content** | **Page.No** |

**2.ABSTRACT**

**OBJECTIVE:**

Now a days many new applications are coming up which are based on online home delivery of groceries. For example we have Big Basket application for delivery of grocery items but we face a problem that we can only buy grocery items from him. We cannot choose the shop from which we want to buy the groceries. In the recent applications like swiggy and zomato which are based on online food delivery we can choose the place from which we can order. This makes the application user friendly. So we have implemented this idea into online grocery delivery. In our application the user can select the shop from which he wants to order his groceries and his groceries will be delivered. The user can now order from his favourite shop instead of buying from the same seller everytime.

**WORKING :**

The code will ask the user his present location. When the user types his location, the shops near to him will be displayed based on the distance from his location. The user can select any of those shops or he can enter a particular shop. When the user enters the shop name, the list of grocery items sold by the seller selected by the user will be displayed along with their prices. The user can add the required items to his cart. Now he can also checkout from that shop and visit another shop as well. He can order from n number of shops at a time. Finally a bill is generated based on the shops he has visited and the orders he has made.

Concepts used: Maps(Shortest path algorithm), Sorting.

Data structure: Maps, LikedLists, Arrays

Language used : C language.

**3.Introduction**

Find my shop is an application through which the users can order their grocery items from their favourite shops online. Now a days many new applications are coming up which are based on online home delivery of groceries. For example we have Big Basket application for delivery of grocery items but we face a problem that we can only buy grocery items from him. We cannot choose the shop from which we want to buy the groceries. In the recent applications like swiggy and zomato which are based on online food delivery we can choose the place from which we can order. This makes the application user friendly. So we have implemented this idea into online grocery delivery. In our application the user can select the shop from which he wants to order his groceries and his groceries will be delivered. The user can now order from his favourite shop instead of buying from the same seller everytime.

The code will ask the user his present location. When the user types his location, the shops near to him will be displayed based on the distance from his location. The user can select any of those shops or he can enter a particular shop. When the user enters the shop name, the list of grocery items sold by the seller selected by the user will be displayed along with their prices. The user can add the required items to his cart. Now he can also checkout from that shop and visit another shop as well. He can order from n number of shops at a time. Finally a bill is generated based on the shops he has visited and the orders he has made.

We have implemented the map of the Hyderabad city with the help of graphs data structure. The graph is implemented with the help of adjacency matrix where the edge length of each edge is stored. To store the details of the items which the user has stored in the cart we have used the linked list data structure. Whenever the user adds an item into his cart the item will be added to the end of the linked list. The items displayed along with their prices in a particular shop is displayed using two dimensional arrays.

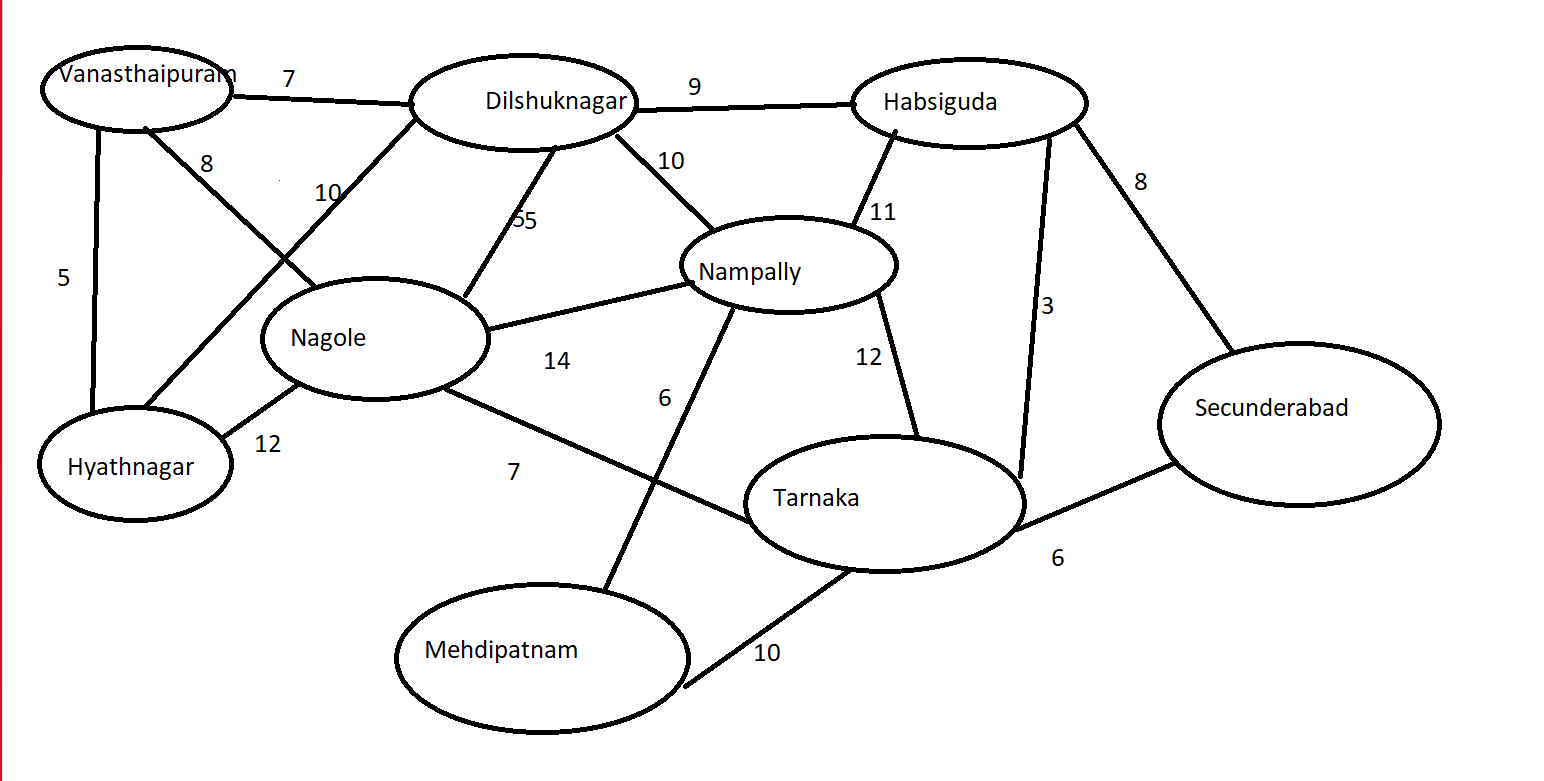
**4. Design**

**4.1 Strategy:**

The code will ask the user his present location. When the user types his location, the shops near to him will be displayed based on the distance from his location. The user can select any of those shops or he can enter a particular shop. When the user enters the shop name, the list of grocery items sold by the seller selected by the user will be displayed along with their prices. The user can add the required items to his cart. Now he can also checkout from that shop and visit another shop as well. He can order from n number of shops at a time. Finally a bill is generated based on the shops he has visited and the orders he has made.

We have implemented the map of the Hyderabad city with the help of graphs data structure. The graph is implemented with the help of adjacency matrix where the edge length of each edge is stored. To store the details of the items which the user has stored in the cart we have used the linked list data structure. Whenever the user adds an item into his cart the item will be added to the end of the linked list. The items displayed along with their prices in a particular shop is displayed using two dimensional arrays.

**4.2 Our Map:**

****

**5. Implementation**

# include <stdio.h>

# include <stdlib.h>

# include <string.h>

# include <time.h>

struct shopName

{

char name[30];

float distance;

struct shopName \*next;

};

struct shopName \*shops[9],\*temp,\*q;char places[9][30];int adj[9][9];char items[22][30];int prices[22];float total=0;

struct shopDetails

{

char name[30];

char place[30];

float distace;

};

struct cartNode

{

char itemName[30];

int quantity;

float price;

float totPrice;

char shopName[30];

struct cartNode \*next;

}\*cart,\*q2,\*temp2;

int getItemPrice(char itemName[])

{

int i;

for(i=0;i<22;i++)

{

if(!strcmp(items[i],itemName))

break;

}

return prices[i];

}

void addToCart(char itemName[],char shopName[],int quantity)

{

temp2=(struct cartNode\*)malloc(sizeof(struct cartNode));

strcpy(temp2->itemName,itemName);

strcpy(temp2->shopName,shopName);

temp2->quantity=quantity;

temp2->price=getItemPrice(itemName);

temp2->totPrice=quantity\*temp2->price;

total+=temp2->totPrice;

temp2->next=NULL;

if(cart==NULL)

{

cart=temp2;

}

else

{

q2=cart;

while(q2->next!=NULL)

{

q2=q2->next;

}

q2->next=temp2;

}

}

struct shopDetails searchShop(char name[])

{

int i;struct shopDetails s;

for(i=0;i<9;i++)

{

q=shops[i];

while(q!=NULL)

{

if(!strcmp(name,q->name))

{

strcpy(s.name,name);

strcpy(s.place,places[i]);

s.distace=q->distance;

return s;

}

q=q->next;

}

}

return s;

}

void displayNearbyShops(int num)

{

int i,count=0;

q=shops[num];

while(q!=NULL)

{

count++;

printf(" %d.%s %.2fkm (%s)\n",count,q->name,q->distance,places[num]);

q=q->next;

}

int min=10000,flag;

for(i=0;i<9;i++)

{

if(adj[num][i]==-1 || adj[num][i]==0)

continue;

if(adj[num][i]<min)

{

min=adj[num][i];

flag=i;

}

}

q=shops[flag];

while(q!=NULL)

{

count++;

printf(" %d.%s %.2fkm (%s)\n",count,q->name,min+q->distance,places[flag]);

if(count==8)

return;

q=q->next;

}

int min2=10000;

for(i=0;i<9;i++)

{

if(adj[num][i]==-1 || adj[num][i]==0 || adj[num][i]==min)

continue;

if(adj[num][i]<min2)

{

min2=adj[num][i];

flag=i;

}

}

q=shops[flag];

while(q!=NULL)

{

count++;

printf(" %d.%s %.2fkm (%s)\n",count,q->name,min2+q->distance,places[flag]);

if(count==8)

return;

q=q->next;

}

}

void displayItems(char shopName[])

{

int j,k;char c;

struct shopDetails s=searchShop(shopName);

printf("\n--------%s (%s)--------\n",s.name,s.place);

int randItems[17]={-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1};

srand(time(0));

for(j=0;j<16;j++)

{

int num = (rand() % (22));

for(k=0;k<j;k++)

{

if(randItems[k]==num)

{

j--;

break;

}

if(k==j-1)

{

randItems[j]=num;

printf(" %d. %-15s Price:-Rs.%d \n",j,items[num],prices[num]);

}

}

}

do

{

char itemName[30];

printf("Enter item name to be added to cart OR Enter 'C' to checkout\n");

scanf("%s",&itemName);

if(!strcmp(itemName,"C"))

return;

else

{

int quantity;

printf("Enter Quantity\n");

scanf("%d",&quantity);

addToCart(itemName,shopName,quantity);

printf("Item added to cart successfully!\n");

}

}while(1);

}

void displayShopsInArea(int num)

{

q=shops[num];int count=1;char shopNameInput[30],areaNameInput[30];

while(q!=NULL)

{

printf(" %d.%s %.2fkm (%s)\n",count,q->name,q->distance,places[num]);

count++;

q=q->next;

}

printf("\nSelect a shop OR Do you want to check shops form other areas?(Enter Other)\n");

scanf(" %[^\n]s",&shopNameInput);

if(!strcmp(shopNameInput,"Other"))

{

char areaNameInput[30];

printf("Enter Area Name..\n");

scanf("%s",&areaNameInput);

int m;

for(m=0;m<9;m++)

{

if(!strcmp(areaNameInput,places[m]))

break;

}

displayShopsInArea(m);

}

else

{

displayItems(shopNameInput);

}

}

void addShopName(int num,char name[],float distance)

{

temp=(struct shopName\*)malloc(sizeof(struct shopName));

strcpy(temp->name,name);

temp->distance=distance;

if(shops[num]==NULL)

{

temp->next=NULL;

shops[num]=temp;

}

else

{

q=shops[num];

if(distance<q->distance)

{

temp->next=shops[num];

shops[num]=temp;

return;

}

while( q->next!=NULL && distance>q->next->distance)

q=q->next;

temp->next=q->next;

q->next=temp;

}

}

int main()

{

strcpy(places[0],"Vanasthalipuram");

strcpy(places[1],"Dilsukhnagar");

strcpy(places[2],"Habsiguda");

strcpy(places[3],"Hayatnagar");

strcpy(places[4],"Nagole");

strcpy(places[5],"Nampally");

strcpy(places[6],"Mehdipatnam");

strcpy(places[7],"Tarnaka");

strcpy(places[8],"Secunderabad");

adj[0][0]=0;

adj[0][1]=7;

adj[0][2]=-1;

adj[0][3]=5;

adj[0][4]=8;

adj[0][5]=-1;

adj[0][6]=-1;

adj[0][7]=-1;

adj[0][8]=-1;

adj[1][0]=7;

adj[1][1]=0;

adj[1][2]=9;

adj[1][3]=10;

adj[1][4]=5;

adj[1][5]=10;

adj[1][6]=-1;

adj[1][7]=-1;

adj[1][8]=-1;

adj[2][0]=-1;

adj[2][1]=9;

adj[2][2]=0;

adj[2][3]=-1;

adj[2][4]=-1;

adj[2][5]=11;

adj[2][6]=-1;

adj[2][7]=3;

adj[2][8]=8;

adj[3][0]=5;

adj[3][1]=10;

adj[3][2]=-1;

adj[3][3]=0;

adj[3][4]=12;

adj[3][5]=-1;

adj[3][6]=-1;

adj[3][7]=-1;

adj[3][8]=-1;

adj[4][0]=8;

adj[4][1]=5;

adj[4][2]=-1;

adj[4][3]=12;

adj[4][4]=0;

adj[4][5]=14;

adj[4][6]=-1;

adj[4][7]=7;

adj[4][8]=-1;

adj[5][0]=-1;

adj[5][1]=10;

adj[5][2]=11;

adj[5][3]=-1;

adj[5][4]=14;

adj[5][5]=0;

adj[5][6]=6;

adj[5][7]=12;

adj[5][8]=-1;

adj[6][0]=-1;

adj[6][1]=-1;

adj[6][2]=-1;

adj[6][3]=-1;

adj[6][4]=-1;

adj[6][5]=6;

adj[6][6]=0;

adj[6][7]=10;

adj[6][8]=-1;

adj[7][0]=-1;

adj[7][1]=-1;

adj[7][2]=3;

adj[7][3]=-1;

adj[7][4]=7;

adj[7][5]=12;

adj[7][6]=10;

adj[7][7]=0;

adj[7][8]=6;

adj[8][0]=-1;

adj[8][1]=-1;

adj[8][2]=8;

adj[8][3]=-1;

adj[8][4]=-1;

adj[8][5]=-1;

adj[8][6]=-1;

adj[8][7]=6;

adj[8][8]=0;

strcpy(items[0],"Sugar");

strcpy(items[1],"Salt");

strcpy(items[2],"Rice");

strcpy(items[3],"Dal");

strcpy(items[4],"Oreo");

strcpy(items[5],"Bourbon");

strcpy(items[6],"Kitkat");

strcpy(items[7],"Munch");

strcpy(items[8],"Lays");

strcpy(items[9],"Doritos");

strcpy(items[10],"Oil");

strcpy(items[11],"Almonds");

strcpy(items[12],"Cashew");

strcpy(items[13],"Anjeer");

strcpy(items[14],"Candles");

strcpy(items[15],"Chilli Powder");

strcpy(items[16],"Turmeric");

strcpy(items[17],"Wheat Flour");

strcpy(items[18],"AXE Deodrant");

strcpy(items[19],"Soaps");

strcpy(items[20],"Shampoo");

strcpy(items[21],"Jeera");

prices[0]=67;

prices[1]=20;

prices[2]=42;

prices[3]=120;

prices[4]=30;

prices[5]=25;

prices[6]=15;

prices[7]=10;

prices[8]=20;

prices[9]=30;

prices[10]=105;

prices[11]=197;

prices[12]=285;

prices[13]=395;

prices[14]=13;

prices[15]=74;

prices[16]=215;

prices[17]=67;

prices[18]=185;

prices[19]=30;

prices[20]=75;

prices[21]=43;

addShopName(0,"Best Food",0.3);

addShopName(0,"Hungry Harvest",0.7);

addShopName(0,"New Natural",0.5);

addShopName(1,"White Sheep Grocery",0.4);

addShopName(1,"Green Gatherings",0.9);

addShopName(1,"Bliss Vista Foods",1.2);

addShopName(1,"Best Hands Food",0.6);

addShopName(2,"Crunchy Bites",0.3);

addShopName(2,"EverGrowth",0.6);

addShopName(3,"Planet Organics",0.8);

addShopName(3,"Gaint Foods",1.1);

addShopName(3,"Happy Mart",0.5);

addShopName(4,"Gold Medal Grocery",0.2);

addShopName(4,"Farmer's Choice",0.9);

addShopName(4,"Rich Roots",1.2);

addShopName(4,"Fresh Land",0.5);

addShopName(5,"Daily Bread",1.3);

addShopName(5,"Tasty Green",0.6);

addShopName(5,"Savvy Market",1.0);

addShopName(6,"Living Market",1.1);

addShopName(6,"Freshly Picked",0.7);

addShopName(6,"PeePod",1.4);

addShopName(7,"Sparrow's Nest",0.4);

addShopName(7,"Infeild",0.7);

addShopName(7,"Insta Cart",0.8);

addShopName(8,"Pirates Of Carrebian",1.2);

addShopName(8,"IronMan Mart",0.5);

printf("Hello User! Welcome to online shopping!!\n");

printf("Enter Your Location....\n");

char placeNameInput[30],shopNameInput[30],c;int i,j,k;

scanf("%s",&placeNameInput);

for(i=0;i<9;i++)

{

if(!strcmp(placeNameInput,places[i]))

break;

}

do

{

printf("\n----------------Shops near you----------------\n");

displayNearbyShops(i);

printf("\nSelect a shop OR Do you want to check shops form other areas?(Enter Other)\n");

scanf(" %[^\n]s",&shopNameInput);

if(!strcmp(shopNameInput,"Other"))

{

char areaNameInput[30];

printf("Enter Area Name..\n");

scanf("%s",&areaNameInput);

int p;

for(p=0;p<9;p++)

{

if(!strcmp(areaNameInput,places[p]))

break;

}

displayShopsInArea(p);

}

else

{

displayItems(shopNameInput);

}

printf("Do you want to buy from other shops?(Y/N)\n");

scanf(" %c",&c);

}while(c!='n'&&c!='N');

q2=cart;

if(q2==NULL)

{

printf("Cart is Empty!!");

}

else

{

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

struct shopDetails shop=searchShop(q2->shopName);

printf("-----------%s (%s)-----------\n",shop.name,shop.place);

printf("%-15s Quantity- %d Price- Rs.%.2f\n",q2->itemName,q2->quantity,q2->totPrice);

while(q2!=NULL)

{

if(q2->next!=NULL)

{

if(strcmp(q2->shopName,q2->next->shopName))

{

shop=searchShop(q2->next->shopName);

printf("\n-----------%s (%s)-----------\n",shop.name,shop.place);

}

}

q2=q2->next;

if(q2==NULL)

break;

printf("%-15s Quantity- %d Price- Rs.%.2f\n",q2->itemName,q2->quantity,q2->totPrice);

}

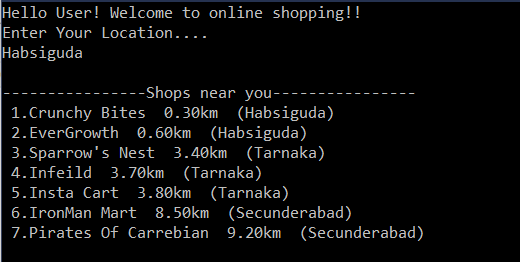
printf("\n TOTAL- Rs.%.2f",total);

}

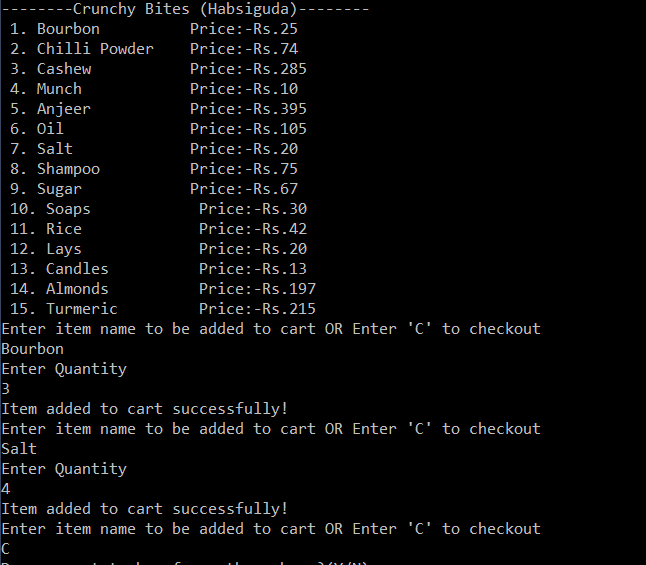
}

**6.OUTPUTS**

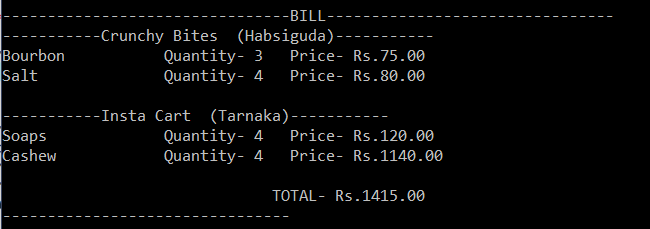
Output 1: When user enters his location shops near him are displayed.



Output 2: User adds items to his cart.



Output 3: Bill is generated.

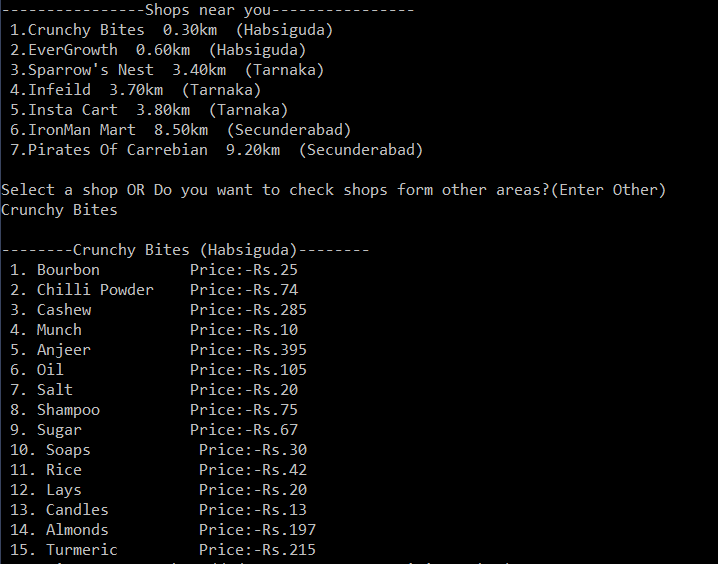


**7.Test cases:**

When user enters his location he is asked to choose one of the two options. He must either enter the shop name which is displayed on the screen or if he wants to choose particular shop then he must enter other. When he enters other he will be asked to give the location of the shop, then the shops in that location are displayed and he can choose his favorite shop.

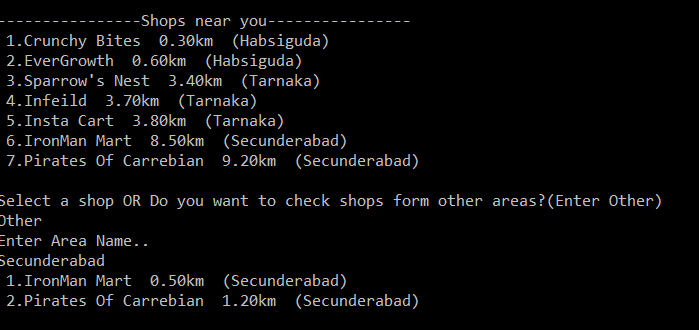
Test case-1.

-> When user enters the shop from the displayed list.



Test case-2:

When the user selects shop from other location.

******

**8.Conclusion**

To conclude, people can use this application to order their grocery items from their favourite shop. Users can also order from two or more shops at a time. After they order a bill is generated and displayed on the screen.

**9.References** :

[1]Algorithm Design text book :Jon Klienberg and Eva tardos

[2 https://www.geeksforgeeks.org/graph-data-structure-and-algorithms/

[3] https://www.tutorialspoint.com/data\_structures\_algorithms/algorithms\_basics.html