**MALNAD COLLEGE OF ENGINEERING, HASSAN**

**[An Autonomous Institution under VTU, Belagavi]**

**Hassan, Karnataka 573202, India**

****

Data Structures Mini-Project Report

On

**“Restaurant Management System”**

Submitted by

**Akash (4MC20CS005)**

**Anvesh.J (4MC20CS017)**

**Ashrith.H(4MC20CS019)**

**B.R.Nikilesh(4MC20CS021)**

**Gagan.H.C(4MC20CS045)**

Under the Guidance of

**SUMANTH C M B.E., M.Tech.**

**Assistant Professor**

**Dept. of Computer Science and Engineering**

**Department of Computer Science and Engineering**

Malnad College of Engineering Hassan

Hassan, Karnataka 573202, India

2021-22

**INTRODUCTION**

The businesses in restaurants are now growing constantly. At the same time, the need for managing its operations and tasks arises. The best way to optimize these activities is growing the business online and automizing it as well.

Today’s generation encourages high-tech services especially over the Internet. The project “Restaurant Management System” is implemented to reduce the manual work and enhances the accuracy of work in a restaurant.

Hence the project is developed proficiently to help restaurant. This system wake to provide service facility to restaurant and owners automate their BILLING OPERATIONS. This system entirely reduces the unnecessary time waste inside the hotel as well as it reduces unnecessary calculation.

RESTAURANT MANAGEMENT SYSTEM project fully developed in ‘C’’ language ‘C’ is a programming language that lets you work quickly and integrate systems more efficiently.

**Programming Language Used :-** C

**Concept Used :-** Doubly Linked List

C is a procedural programming language. It was initially developed by Dennis Ritchie in the year 1972. It was mainly developed as a system programming language to write an operating system.

The main features of the C language include –

- low-level memory access

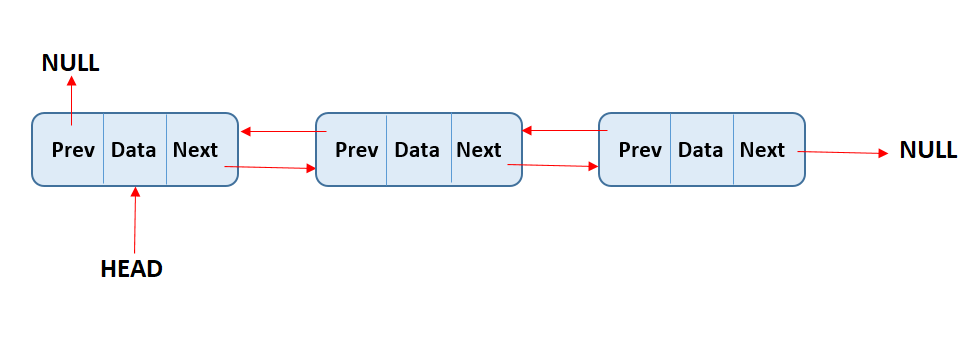
- a simple set of keywords

- clean style

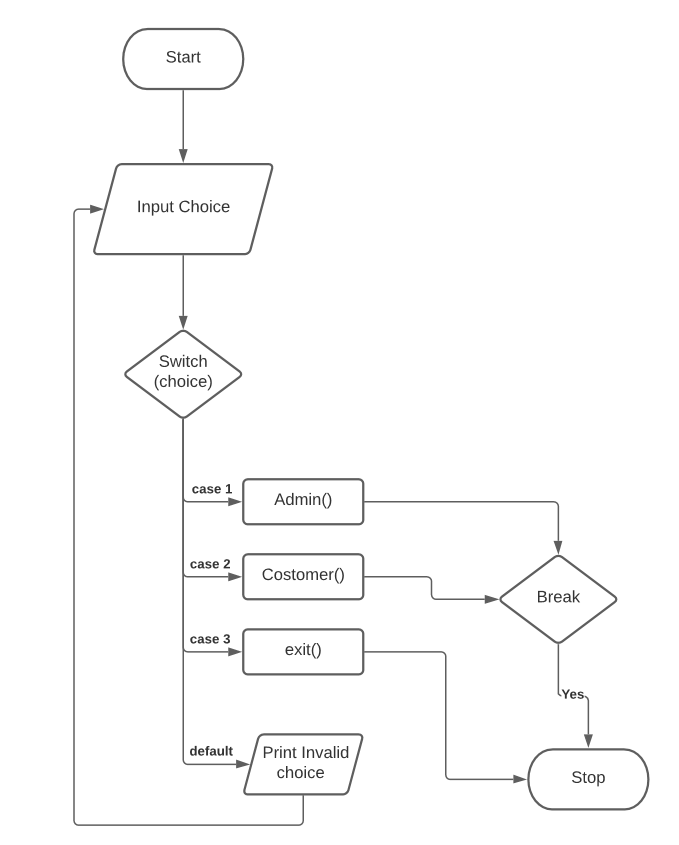
these features make C language suitable for system programming like an operating system or compiler development.

**SYSTEM DESIGN**

Here we mainly use doubly linked list as to store in memory, as they are more efficient than arrays. Unlike arrays, the size for a linked list is not pre-defined, allowing the linked list to increase or decrease in size as the program runs. This is possible because to insert or delete from a linked list, the pointers need to be updated accordingly.



**FLOWCHART**

****

**IMPLEMENTATION**

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

struct node

{ char foodname[100];

int quantity;

float price;

int data;

struct node \*llink;

struct node \*rlink;

};

typedef struct node \*NODE;

NODE firstc=NULL; NODE curc=NULL;NODE newnode;

NODE firsta=NULL; NODE cura=NULL;NODE head\_s;

NODE createadmin(NODE head,int data, char foodname[50], float price) //Add new item to the order list

{ newnode = (NODE)malloc(sizeof(struct node));

newnode->data = data;

newnode->price = price;

newnode->quantity = 0;

strcpy(newnode->foodname,foodname);

newnode->rlink = NULL;

newnode->llink = NULL;

NODE temp = head;

if(temp==NULL)

{ cura = newnode;

firsta = cura;

}

else

{

while(temp->rlink!=NULL)

{ temp=temp->rlink;

}

temp->rlink=newnode;

newnode->llink = cura;

cura = newnode;

}

return firsta;

}

NODE createcustomer(NODE head,int data,int quantity) // order the item

{

newnode = (NODE)malloc(sizeof(struct node));

NODE temp1 = firsta;

int flag = 0;

while(temp1!=NULL)

{

if(temp1->data==data)

{ flag = 1;

break;

}

temp1 = temp1->rlink;

}

if(flag==1)

{

newnode->data = data;

newnode->price = quantity\*(temp1->price);

newnode-> quantity = quantity;

strcpy(newnode->foodname,temp1->foodname);

newnode->rlink = NULL;

newnode->llink = NULL;

NODE temp = head;

if(temp==NULL)

firstc = curc = newnode;

else

{

while(temp->rlink!=NULL)

temp=temp->rlink;

temp->rlink=newnode;

newnode->llink = curc;

curc = newnode;

}

}

else

{

printf("\n\t\t\t\t\t\t\tThis item is not present in the menu!\n");

}

return firstc;

}

void displayList(NODE head) // display the list of items

{

NODE temp1 = head;

if(temp1==NULL)

{

printf("\n\t\t\t\t\t\t\t\tList is empty!!\n\n");

}

else

{

printf("\n");

while(temp1!=NULL)

{

if(temp1->quantity==0)

printf("\t\t\t\t\t\t\t%d\t%s\t%0.2f\n",temp1->data,temp1->foodname,temp1->price);

else

{

printf("\t\t\t\t\t\t\t%d\t%s\t%d\t%0.2f\n",temp1->data,temp1->foodname,temp1->quantity,temp1->price);

}

temp1 = temp1->rlink;

}

printf("\n");

}

}

NODE totalsales(int data,int quantity) // view total sales

{

newnode = (NODE)malloc(sizeof(struct node));

int flag = 0;

NODE temp1 = firsta;

while(temp1->data!=data)

{

temp1 = temp1->rlink;

}

newnode->data = data;

newnode->price = quantity\*(temp1->price);

newnode-> quantity = quantity;

strcpy(newnode->foodname,temp1->foodname);

newnode->rlink = NULL;

newnode->llink = NULL;

NODE temp = head\_s;

if(temp==NULL)

head\_s = newnode;

else

{

while(temp->rlink!=NULL)

{

if(temp->data==data)

{

flag = 1;

break;

}

temp=temp->rlink;

}

if(flag==1)

{

temp->quantity += newnode-> quantity;

temp->price += newnode->price;

}

else

{

temp->rlink=newnode;

}

}

return head\_s;

}

void calculatetotsales() // calculate

{

NODE temp = firstc;

while(temp!=NULL)

{

head\_s = totalsales(temp->data, temp->quantity);

temp=temp->rlink;

}

}

NODE delete(int data,NODE head, NODE tail) // delete item from menu or

{

if(head==NULL)

{

printf("\n\t\t\t\t\t\t\tList is empty\n");

}

else

{

NODE temp;

if(data==head->data)

{

temp = head;

head = head->rlink;

if (head != NULL)

head->llink = NULL;

free(temp);

}

else if(data==tail->data)

{

temp = tail;

tail = tail->llink;

tail->rlink = NULL;

free(temp);

}

else

{

temp = head;

while(data!=temp->data)

{

temp = temp->rlink;

}

(temp->llink)->rlink = temp->rlink;

(temp->rlink)->llink = temp->llink;

free(temp);

}

}

return head;

}

int deleteadmin() // delete item from order menu

{

printf("\n\t\t\t\t\tEnter serial no. of the food item which is to be deleted: ");

int num;

scanf("%d",&num);

NODE temp=firsta;

while(temp!=NULL)

{

if (temp->data == num)

{

firsta = delete(num, firsta, cura);

return 1;

}

temp=temp->rlink;

}

return 0;

}

int deletecustomer() //delete item from the order

{

printf("\n\t\t\t\t\tEnter serial no. of the food item which is to be deleted: ");

int num;

scanf("%d",&num);

NODE temp=firstc;

while(temp!=NULL)

{

if (temp->data == num)

{

firstc = delete(num, firstc, curc);

return 1;

}

temp=temp->rlink;

}

return 0;

}

void displaybill(NODE head) // display bill

{

NODE temp = head;

float total\_price = 0;

while (temp!=NULL)

{

total\_price +=temp->price;

temp = temp->rlink;

}

printf("\t\t\t\t\t\t\tTotal price: %0.02f\n",total\_price);

}

NODE deleteList(NODE head) //delete node

{

if(head==NULL)

{

return NULL;

}

else

{

NODE temp = head;

while(temp->rlink!=0)

{

temp = temp->rlink;

free(temp->llink);

}

free(temp);

head = NULL;

}

return head;

}

void admin()

{ int opt,pass;

printf("\n\t\t\t\t\t\tEnter password : ");

scanf("%d",&pass);

if(pass==12345678)

{

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

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

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

do

{

printf("\n\t\t\t\t\t\t\t1. View total sales\n");

printf("\t\t\t\t\t\t\t2. Add new items in the order menu\n");

printf("\t\t\t\t\t\t\t3. Delete items from the order menu\n");

printf("\t\t\t\t\t\t\t4. Display order menu\n");

printf("\t\t\t\t\t\t\t5. Back To Main Menu \n\n");

printf("\t\t\t\t\t\t\t Enter Your Choice --->");

scanf("%d",&opt);

switch (opt)

{

case 1:

displayList(head\_s);

displaybill(head\_s);

break;

case 2:

printf("\n\t\t\t\t\t\t\tEnter serial no. of the food item: ");

int num,flag = 0;

char name[50];

float price;

scanf("%d",&num);

NODE temp = firsta;

while(temp!=NULL)

{

if(temp->data==num)

{

printf("\n\t\t\t\t\t\tFood item with given serial number already exists!!\n\n");

flag = 1;

break;

}

temp = temp->rlink;

}

if(flag==1)

break;

printf("\t\t\t\t\t\t\tEnter food item name: ");

scanf("%s",name);

printf("\t\t\t\t\t\t\tEnter price: ");

scanf("%f",&price);

firsta = createadmin(firsta, num, name, price);

printf("\n\t\t\t\t\t\t\tNew food item added to the list!!\n\n");

break;

case 3:

if(deleteadmin())

{

printf("\n\t\t\t\t\t\t### Updated list of food items menu ###\n");

displayList(firsta);

}

else

printf("\n\t\t\t\t\t\tFood item with given serial number doesn't exist!\n\n");

break;

case 4:

printf("\n\t\t\t\t\t\t\t ### Order menu ###\n");

displayList(firsta);

break;

case 5:

break;

}

}while(opt!=5);

}

else

{ printf("\n\t\t\t\t\t\tWrong Password\n\n");

}

}

void customer()

{

int opt,flag=0;

char ch;

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

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

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

do

{

printf("\n\t\t\t\t\t\t\t1. Place your order\n");

printf("\t\t\t\t\t\t\t2. View your ordered items\n");

printf("\t\t\t\t\t\t\t3. Delete an item from order\n");

printf("\t\t\t\t\t\t\t4. Display final bill\n");

printf("\t\t\t\t\t\t\t5. Back To Main Menu \n\n");

printf("\t\t\t\t\t\t\t Enter Your Choice --->");

scanf("%d",&opt);

switch (opt)

{

case 1:

displayList(firsta);

printf("\n\t\t\t\t\t\tEnter number corresponding to the item you want to order: ");

int n;

scanf("%d",&n);

printf("\t\t\t\t\t\tEnter quantity: ");

int quantity;

scanf("%d",&quantity);

firstc = createcustomer(firstc, n, quantity);

printf("\n\t\t\t\t\t\tOrder Placed !!\n");

break;

case 2:

printf("\n\t\t\t\t\t\t\t ### List of ordered items ###\n");

displayList(firstc);

break;

case 3:

if(deletecustomer())

{

printf("\n\t\t\t\t\t\t### Updated list of your ordered food items ###\n");

displayList(firstc);

}

else

printf("\n\t\t\t\t\t\tFood item with given serial number doesn't exist!!\n");

break;

case 4:

calculatetotsales();

printf("\n\t\t\t\t\t\t\t ### Final Bill ###\n");

displayList(firstc);

displaybill(firstc);

firstc = deleteList(firstc);

break;

case 5 :

break;

}

}while(opt!=5);

}

void main()

{ int choice;

firsta = createadmin(firsta,1,"TomatoSoup",60.0);

firsta = createadmin(firsta,2,"GobiManchurian",120.0);

firsta = createadmin(firsta,3,"Noodles",120.0);

firsta = createadmin(firsta,4,"FriedRice",160.0);

firsta = createadmin(firsta,5,"Parota",50.0);

do

{

printf("\n \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf(" WELCOME TO RESTAURANT MANAGEMENT SYSTEM\n");

printf(" \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n\n\n");

printf("\t\t\t\t\t\t\t1. ADMIN SECTION--> \n");

printf("\t\t\t\t\t\t\t2. CUSTOMER SECTION--> \n");

printf("\t\t\t\t\t\t\t3. Exit--> \n\n");

printf("\t\t\t\t\t\t\tEnter Your Choice --->");

scanf("%d",&choice);

switch (choice)

{

case 1: admin();

break;

case 2: customer();

break;

case 3: printf("\t\t\t\t\t\t\tTHANK YOU\n");

exit(0);

break;

default:

printf("\n\t\t\t\t\t\tWrong Input !! Please choose valid option\n");

break;

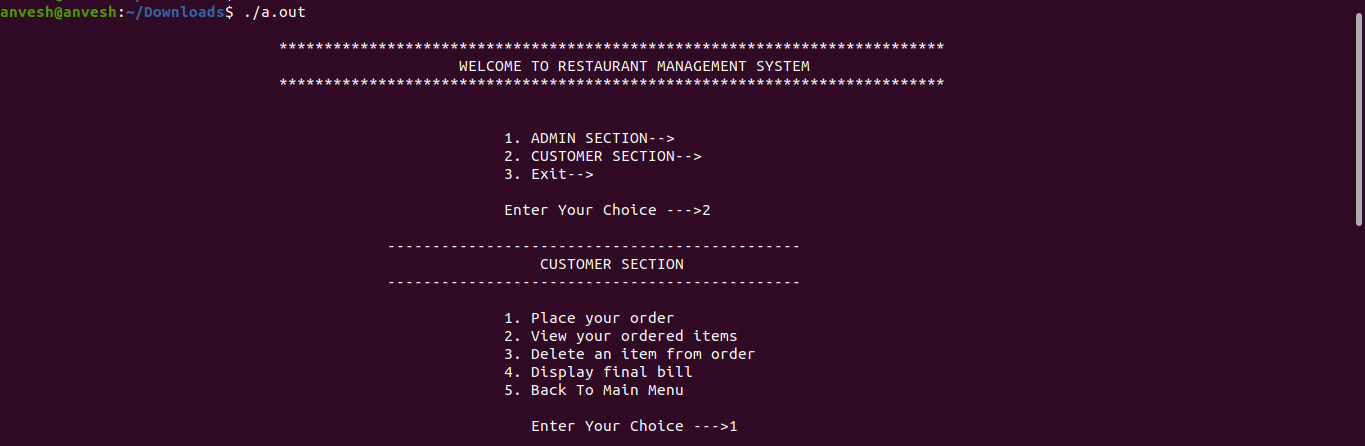
}

}while(choice>0);

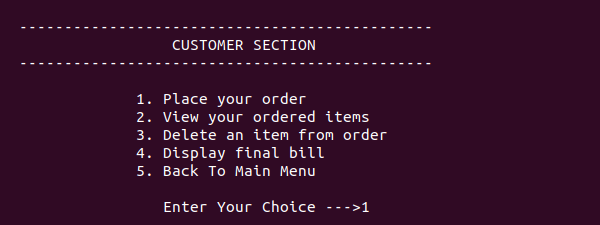
}

**RESULTS**

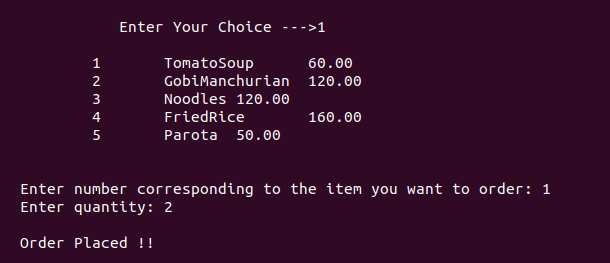
**Home Page layout**

****

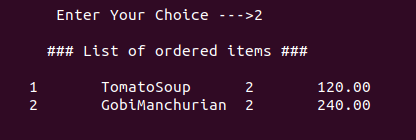
**Customer Section**



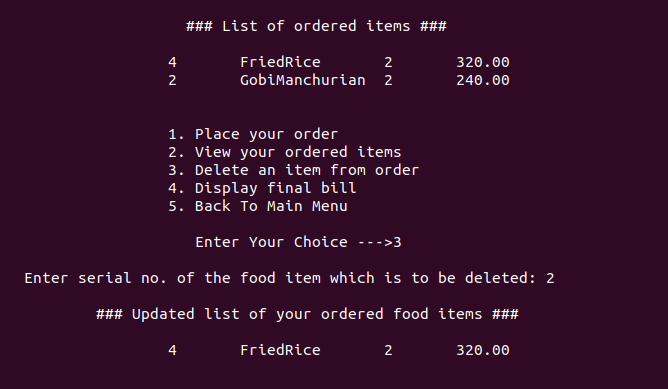
**Placing order**



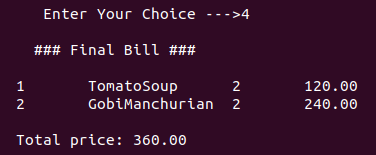
**Viewing ordered items**



**Delete an item from order**

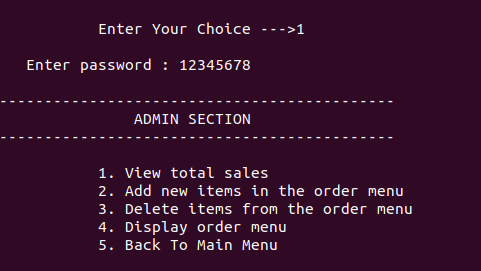
****

**Displaying final bill**

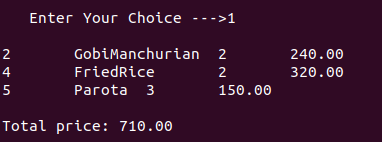


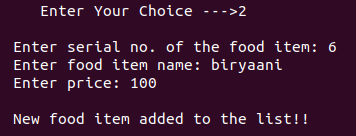
**ADMIN SECTION**

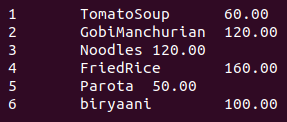
**Password protected admin section**

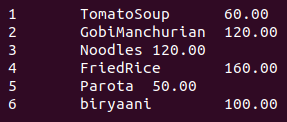


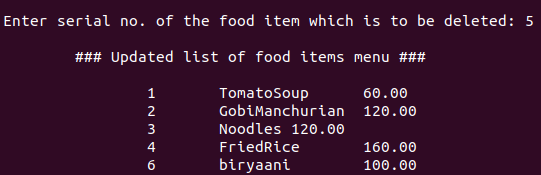
**Viewing total sales**



**Adding new items**



**Deleting item from order**



**REFERENCES: https://youtu.be/chmJqfjQ4m4**