**DS Mini Project Documentation**

**Name: Tanmay Bhor, Krish Devrukhkar, Om Dhumal, Suyash Dhumal**

**Roll No.: 07, 16, 18, 19**

1. **Title of Project: CAFETERIA MANAGEMENT SYSTEM**

**2.**      **Problem Statement:**

This system is made on the demand of managing the offline transaction system better by introducing a management system that works on the user’s personal device such as a computer, and helps people with placing their order through this system and even helps the administrator access the system to look into the transactions.

1. **Objectives:**

Cafeteria Management system is not only aimed to reduce the waiting time of the customer, but also to reduce the manual upkeep required from the coordinator.

The interface would allow the Admin to log in using special credentials. On his dashboard, he can view all the details regarding the customer transactions and the order placed throughout the day. Also, he can view the total money earned for the day. The coordinator can also add and update inventory items, and their price through the admin console present in the system.

1. **Coding:**

#include<iostream.h>

#include<stdlib.h>

#include<conio.h>

#include<string.h>

struct Item

{

char name[50];

int rating;

int price;

int food\_Id\_No;

struct Item \*next;

struct Item \*prev;

};

int cust\_id=1;

struct Item\* head;

struct Item\* last;

int Today\_custmer=0;

int total\_income=0;

struct order\_hist

{

int Customer\_ID;

int items[10][2];

int amount;

char date[11];

struct order\_hist \*next;

struct order\_hist \*prev;

};

struct order\_hist\* head1;

struct order\_hist\* last1;

struct Item\* getnewNode(char a[],int p,int fin)

{

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

temp->food\_Id\_No=fin;

strcpy(temp->name,a);

temp->rating=4;

temp->price=p;

temp->next=NULL;

temp->prev=NULL;

return temp;

}

void insert(char n[],int p,int fin)

{

struct Item\* temp1=getnewNode(n,p,fin);

if(head==NULL)

{

head=temp1;

last=temp1;

}

else

{

temp1->prev=last;

last->next=temp1;

last=temp1;

}

}

struct order\_hist\* getnewNode\_hist()

{

struct order\_hist\* temp=(struct order\_hist\*)malloc(sizeof(struct order\_hist));

temp->next=temp->prev=NULL;

return (temp);

}

void Display()

{

int price,rating;

cout<<" ------------- \n";

cout<<"---------------------------------MENU-------------------------------------------------\n";

cout<<" ------------- \n";

cout<<"INDEX ITEM NAME PRICE RATING\n";

struct Item\* temp=head;

if(head==NULL)

{

cout<<"Empty";

}

while(temp!=NULL)

{

cout<<"\n"<<temp->food\_Id\_No<<"\t\t"<<temp->name<<"\t\t"<<temp->price<<"\t\t"<<temp->rating<<"\n";

temp=temp->next;

}

cout<<"--------------------------------------------------------------------------------\n";

}

int login()

{

char username[20];

char userpwd[11]; // for storing password

int i;

cout<<"Enter your User Name : ";

cin>>username;

cout<<"Enter your password : ";

for(i=0;i<10;i++)

{

userpwd[i]=getch();

cout<<"\*";

}

userpwd[i]='\0';

if(!strcmp(username,"admin") && !strcmp(userpwd,"admin12345"))

{

cout<<"\n\nLogged In Successful\n";

return 1;

}

else

{

cout<<"\n\nIncorrect username or password\n";

return 0;

}

}

void order()

{

int a[10][2];

int n,j=0,i=0;

do{

cout<<"Please enter the FOOD ID NUMBER OF ITEM AND ITS QUANTITY\nEnter input:";

for(i=0;i<2;i++)

{

cin>>a[j][i];

}

j++;

cout<<"You want more items\n1.Yes\n2.No\nEnter input:";

cin>>n;

}while(n==1);

int total\_amount=0;

char name[25];

char Date[10];

cout<<"Enter your Name : ";

cin>> name;

cout<<"Enter Today's Date : ";

cin>>Date;

int k=0;

cout<<"\n---------------------------------------------------------------------------\n";

cout<<" BILL \n";

cout<<"\n---------------------------------------------------------------------------\n";

cout<<"\nName:"<< name;

cout<<"\nDate:"<< Date<<"\n";

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

{

struct Item\* temp=head;

while(temp->food\_Id\_No!=a[k][0])

{

temp=temp->next;

}

cout<<"--------------------------------------------------------------------------------\n";

cout<<"ID:"<<temp->food\_Id\_No<<"\tItem Name: "<<temp->name<<"\tQ:"<<a[k][1]<<"\t"<<"P:"<<(a[k][1]\*(temp->price))<<"\n";

total\_amount+=(a[k][1]\*(temp->price));

}

cout<<"----------------------------------------------------------------------------\n";

cout<<"\nTotal Payable amount is: "<<total\_amount<<"\n";

cout<<"\n--------------------------------------------------------------------------------\n";

struct order\_hist\* temp2=getnewNode\_hist();

temp2->amount=total\_amount;

temp2->Customer\_ID=cust\_id++;

int p,s;

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

{

for(s=0;s<2;s++)

{

temp2->items[p][s]=a[p][s];

}

}

if(head1==NULL)

{

head1=last1=temp2;

}

else

{

last1->next=temp2;

temp2->prev=last1;

last1=temp2;

}

strcpy(temp2->date,Date);

Today\_custmer++;

total\_income+=total\_amount;

}

void display\_rd\_hist()

{

cout<<" ------------- ";

cout<<"\n---------------------------------------ORDER HISTORY-------------------------------------------------";

cout<<" ------------- \n";

cout<<"SR\_NO DATE TOTAL AMOUNT\n";

struct order\_hist\* temp=head1;

if(head1==NULL)

{

cout<<"Empty";

}

while(temp!=NULL)

{

cout<<"\t\n"<<temp->Customer\_ID<<"\t"<<temp->date<<"\t"<<temp->amount;

temp=temp->next;

}

}

void main()

{

clrscr();

head=NULL;

last=NULL;

insert("Burger ",70,1);

insert("Pizza ",90,2);

insert("Hot Cake ",75,3);

insert("Coffee ",70,4);

insert("Ice-Cream",70,5);

insert("Sandwich ",60,6);

insert("Grill ",52,7);

insert("Nun-Bread",35,8);

insert("Cold Drinks",20,9);

int choice;

do{

cout<<"\n------------------------------------------------------------------------------";

cout<<"\n1....FOOD PART";

cout<<"\n2....ADMIN PANEL";

cout<<"\n3....EXIT";

cout<<"\nEnter your choice:";

cin>>choice;

switch(choice)

{

case 1:

cout<<"";

int ex;

do{

cout<<"1.FOOD LIST\n2.ORDER\nEnter Input:";

int n;

cin>> n;

switch(n)

{

case 1: Display();

break;

case 2: order();

break;

}

cout<<"1.FOOD PART\n2.MAIN MENU\nEnter input:";

cin>> ex;

}while(ex==1);

break;

case 2:

cout<<"\n";

int k=login();

int e;

if(k==1){

do{

cout<<"--------------------------------------------------------\n";

cout<<"1. ADD NEW DISH\n";

cout<<"2. TODAY'S TOTAL INCOME\n";

cout<<"3. TODAY TOTAL NO OF CUSTOMER\n";

cout<<"4. ORDER HISTORY\n";

cout<<"5 LIST OF ITEM\n";

cout<<"\n------------------------------------------------------\n";

int ch;

cin>> ch;

switch(ch)

{

case 1:

cout<<" ";

float p;

int fin;

char n[50];

cout<<"Enter the name of Item: ";

cin>> n;

cout<<"Enter the Price of Item: ";

cin>> p;

cout<<"Enter the Food Id of Item: ";

cin>> fin;

insert(n,p,fin);

cout<<"--------------------------------------------------------------------------------\n";

cout<<" NEW DISH IS ADDED SUCESSFULLY.......\n";

cout<<"--------------------------------------------------------------------------------\n";

cout<<"\n";

break;

case 2:

cout<<"------------------------------------------------------------------------------\n";

cout<<" TODAY'S TOTAL INCOME IS:- "<<total\_income;

cout<<"\n------------------------------------------------------------------------------\n";

break;

case 3:

cout<<"------------------------------------------------------------------------------\n";

cout<<" TODAY TOTAL NO OF CUSTOMER VISITED:-"<<Today\_custmer;

cout<<"\n------------------------------------------------------------------------------\n";

break;

case 4:

cout<<"------------------------------------------------------------------------\n";

cout<<" ORDER SUMMARY";

cout<<"\n------------------------------------------------------------------------\n";

display\_rd\_hist();

cout<<"\n";

break;

case 5:

cout<<"\n";

Display();

break;

}

cout<<"1.ADMIN PANEL\n2.MAIN MENU\n";

cin>> e;

}while(e==1);

}

break;

}

}

while(choice!=3);

getch();

}

1. **Screenshots:**

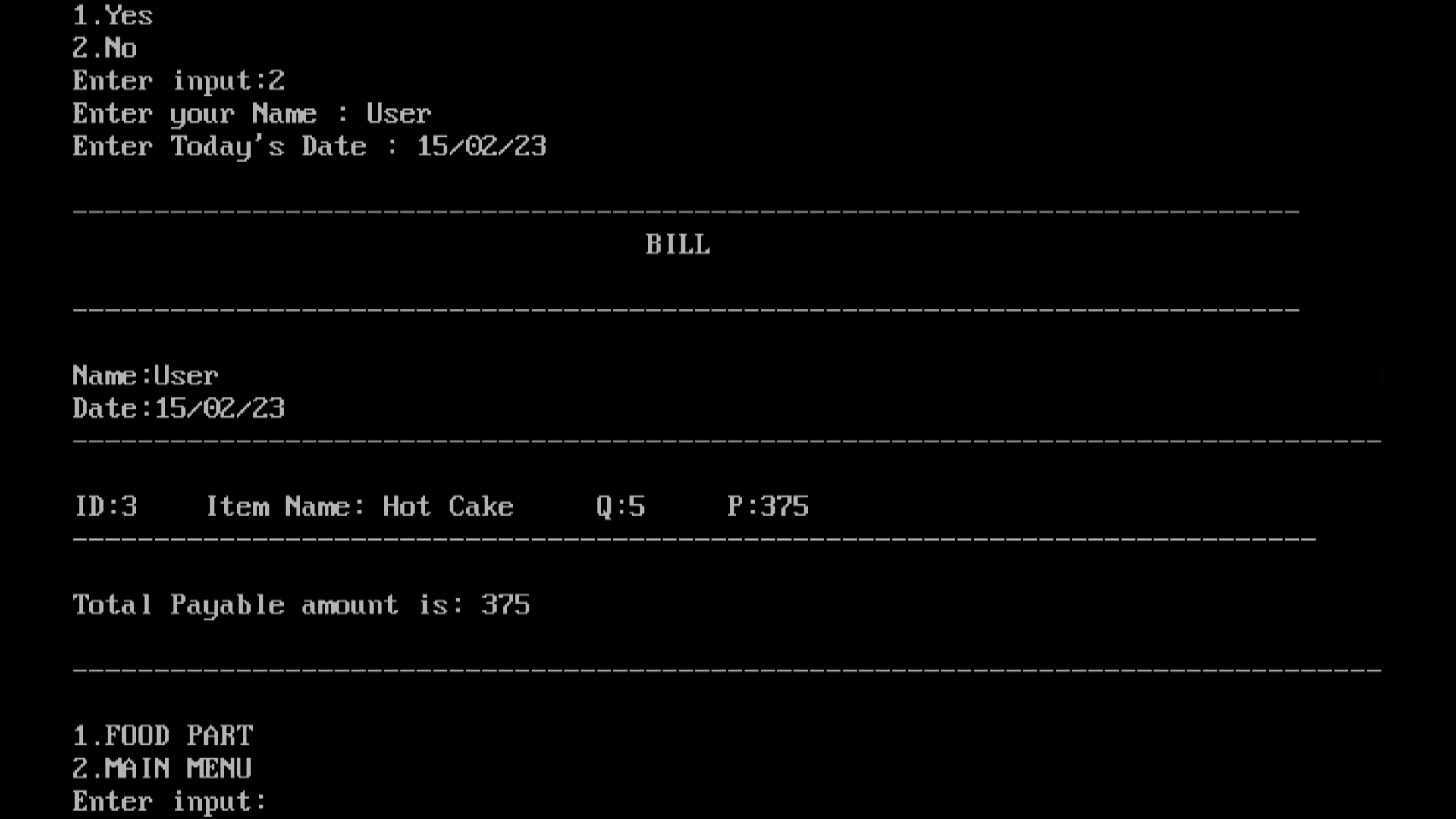
**User Page:**



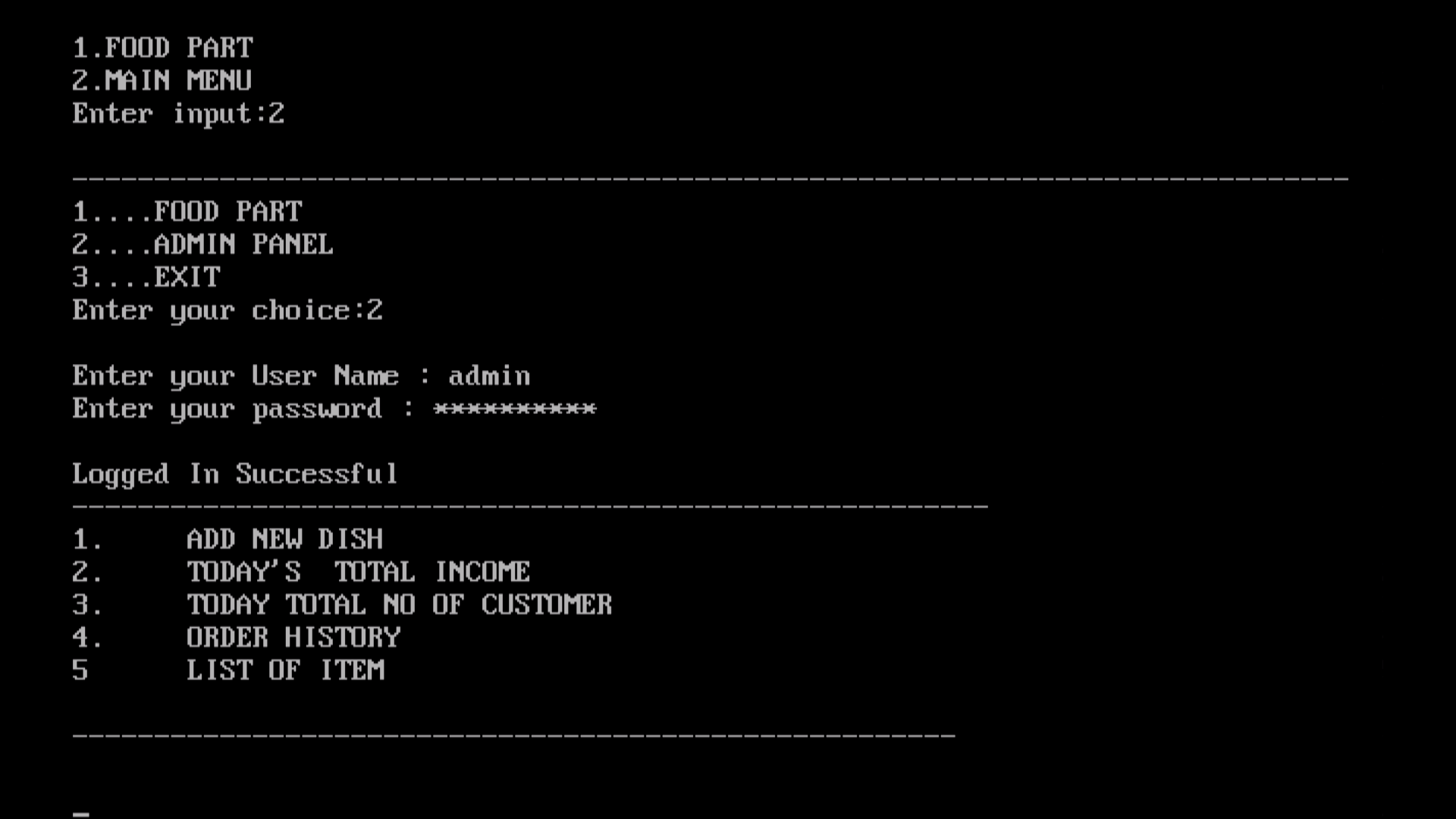
**Café Menu:**



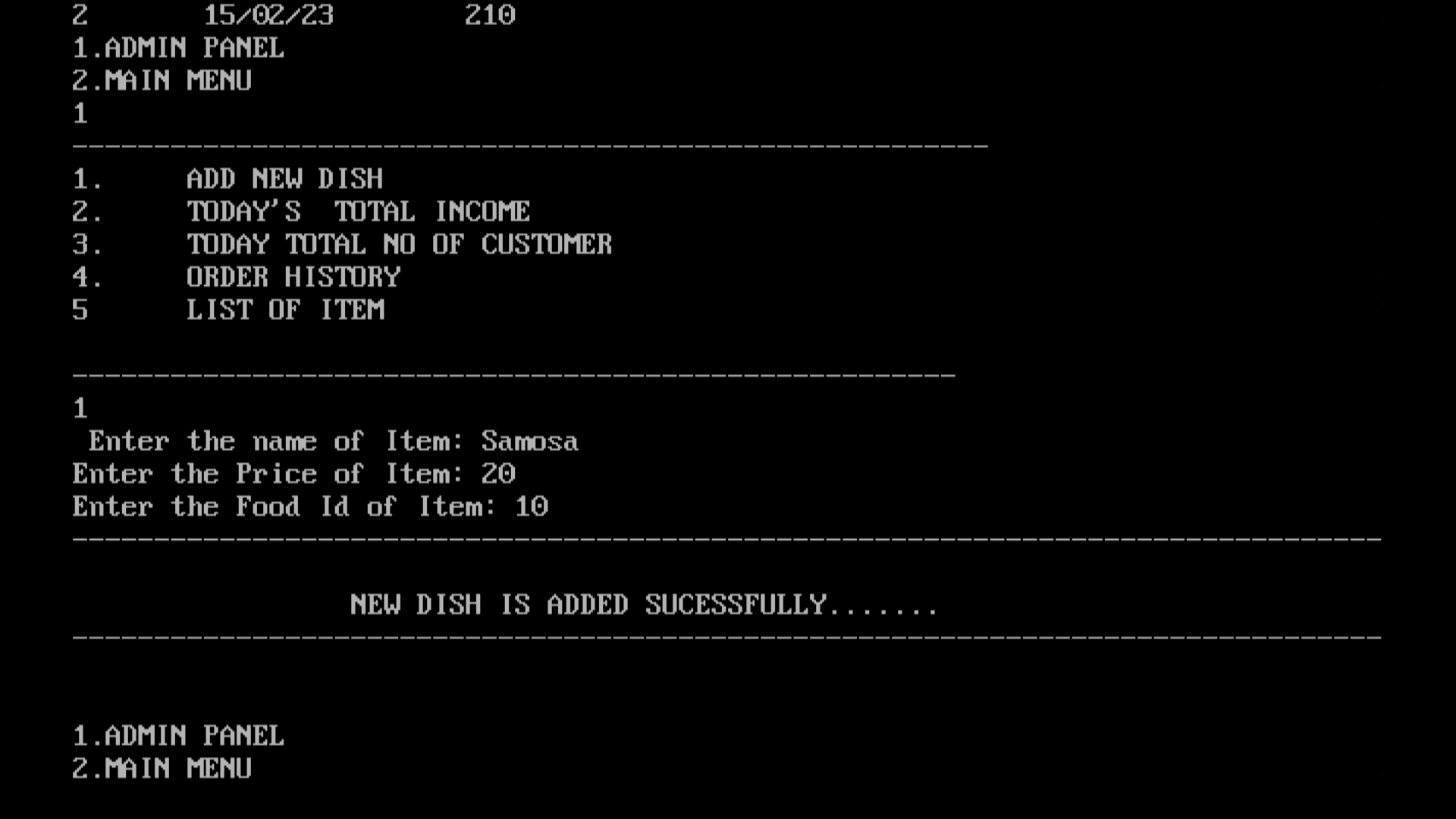
**Ordering Page:**



**Admin Page:**



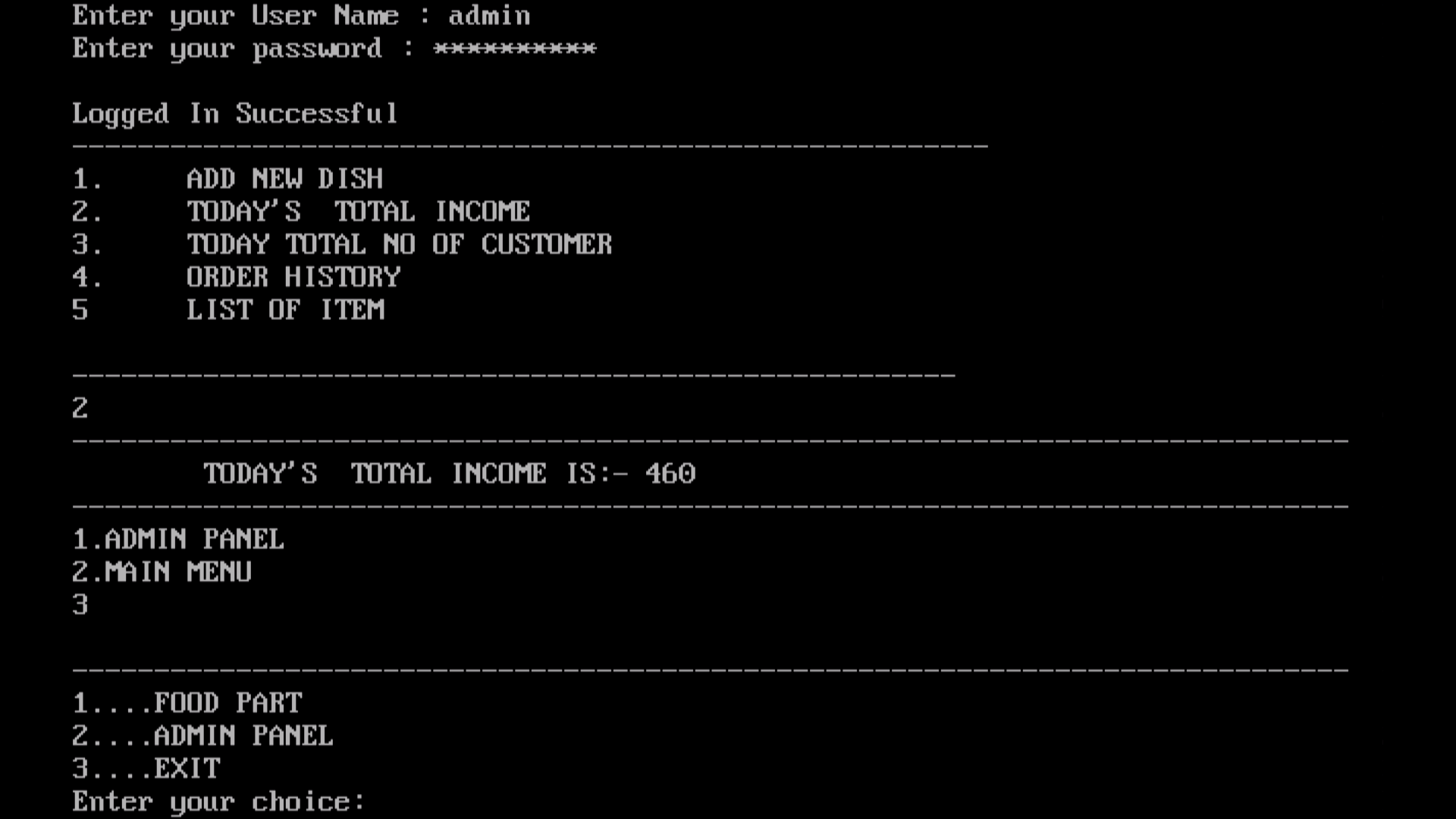
**Adding new item in menu:**



**Item added successfully:**



**Information about new customer details:**



**History of customers who ordered food:**

