

HOTEL MANAGEMENT SYSTEM

NAME::

SHAIK ISMIYA – RA2111047010148

ABHI RAGHAVA –RA2111047010146

SUBJECT :: Moving on, A Hotel Management System In C is untroublesome as it will serve the admin or user to be updated about the records without any strain and it is favored much by the people involved in the business sector. As we are aware of the busy and hectic schedule of business people, this Hotel Management System In C Language turns out to be a great relief for them.

What are hotel management systems?

—> A hotel management system is a set of hotel software solutions that keep operations flowing. There are accounting packages, customer relationship management (CRM) packages, and a dizzying array of industry-specific software.

What are the benefits of hotel?

—→24-hour reception and room service are just a phone call away.

Uniformed security patrol the hotel 24-hour a day, seven days a week to protect guests. Our guests enjoy on-site entertainment such as the video arcade game rooms and Ozzie's Splash Zone water playground.

Source Code ::

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

int itm=1,token=0,total=0,waiting=0,table=0;
int q[5],front=-1,rear=-1;
// Structure for Menu items
struct Menu{
    int item_no;
    char name[20];
    int price;
    struct Menu *next;
} *start1;

// Structure for Token no
struct User{
    int token_no;
    int count;
    int item_id[10];
    struct User *nextToken;
} *start2;

//-----ADMIN-----
// To insert an item in Menu
void create_Menu()
{
    struct Menu *temp, *newnode;
    int val;
    char item[20];

    printf("\n\tADDING AN ITEM\n");
    printf("\tItem no : %d\n",itm);
    printf("\tEnter the name of item : ");
```

```

scanf("%s",item);
printf("\tEnter the price of item : ");
scanf("%d",&val);
newnode = (struct Menu *)malloc(sizeof(struct Menu));
newnode->item_no = itm;
strcpy(newnode->name,item);
newnode->price = val;
newnode->next = NULL;

if(start1 == NULL)
    start1 = newnode;
else {
    temp = start1;
    while(temp->next != NULL)
        temp = temp->next;
    temp->next = newnode;
}
++itm;
}
//-----CUSTOMER-----
// To delete an item in Menu
void delete_Menu()
{
    struct Menu *temp, *prev;
    int no;
    printf("\tREMOVING AN ITEM\n");
    if(start1 == NULL)
        printf("\tMenu is Empty\n");
    else {
        printf("\tEnter the item no : ");
        scanf("%d",&no);
        temp = start1;
        while(temp != NULL && temp->item_no != no) {
            prev = temp;
            temp = temp->next;
        }
    }
}

```

```

        if(temp == start1)
            start1 = temp->next;
        else if(temp != NULL)
            prev->next = temp->next;

        if(temp == NULL)
            printf("\tItem does not exist\n");
        else
            free(temp);
    }
}

// To display the item in Menu
void display_Menu()
{
    struct Menu *temp;
    if(start1 == NULL)
        printf("\tMenu is Empty\n");
    else {
        temp = start1;
        printf("*****MENU*****\n");
        printf("\tItem No\t\tItem Name\t\tItem Price\n");
        while(temp != NULL) {

printf("\t%d\t\t%-20s\t\t%d\n",temp->item_no,temp->name,temp->price);
            temp = temp->next;
        }
    }
}

/////-----Waiting-----
void insert_Wait(int token_id)
{
    if(front == (rear + 1)%5)
        printf("Waiting list full\n");
    else if(rear == -1)

```

```
{
    front = rear = 0;
    q[rear] = token_id;
}
else
{
    rear = (rear + 1)%5;
    q[rear] = token_id;
}
}
```

```
int remove_Wait()
{
    int val;
    if(front == rear)
    {
        val = q[front];
        front = rear = -1;
    }
    else
    {
        val = q[front];
        front = (front + 1)%5;
    }
    return val;
}
```

```
void display_Wait()
{
    int i;
    if(rear == -1)
        printf("\tCurrently Empty\n");
    else
    {
        printf("<-----Waiting List----->\n");
        printf("\tToken No\n");
    }
}
```

```

        for(i = front; i != rear; i = (i+1)%5)
            printf("\t%d\n",q[i]);
        printf("\t%d\n",q[i]);
    }
}

//-----CUSTOMER-----
// To create a new token for the customer
void create_Token(int token_id)
{
    struct User *newnode,*temp;
    int val=0,count=0;
    newnode = (struct User *)malloc(sizeof(struct User));
    newnode->token_no = token_id;
    newnode->nextToken = NULL;
    if(table < 5)
    {
        ++table;
        printf("Table No: %d\n",table);
        display_Menu();
        printf("\tToken no: %d\n",token);
        printf("\tEnter the no of items you wish to add (Press -1 to
submit)\n");
        scanf("%d",&val);
        while(val != -1 && val <= itm)
        {
            newnode->item_id[++count] = val;
            scanf("%d",&val);
        }
        newnode->count = count;
        if(start2 == NULL)
            start2 = newnode;
        else
        {
            temp = start2;
            while(temp->nextToken != NULL)

```

```

        temp = temp->nextToken;
        temp->nextToken = newnode;
    }
}

```

// To delete the token whenever a user checks out

```
void checkout_Token()
```

```

{
    struct User *temp;
    struct Menu *temp2;
    int total_price,count,token_id;
    //count = temp->count;
    total_price = 0;
    if(start2 == NULL)
        printf("No Table is occupied\n");
    else
    {
        temp = start2;
        printf("\tEnter the token id: ");
        scanf("%d",&token_id);
        while(temp != NULL && temp->token_no != token_id)
            temp = temp->nextToken;
        if(temp == NULL)
            printf("\tToken No does not exist\n");
        else
        {
            count = temp->count;
            printf("Item No\tItem Name\tItem Price\n");
            for(int i=1; i<=count; i++)
            {
                temp2 = start1;
                while(temp2 != NULL && temp2->item_no != temp->item_id[i])
                    temp2 = temp2->next;

```

```

                printf("%d\t%s\t%d\n",temp2->item_no,temp2->name,temp2->price);
            }
        }
    }
}

```

```

        total_price += temp2->price;
    }
    printf("Total: %d\n",total_price);
    total += total_price;
    --table;
}
if(temp == start2)
    start2 = start2->nextToken;
free(temp);
}
}

int main()
{
    int ad,ch=0,ch1=0,ch2=0;
    printf("<-----WELCOME TO THE FOOD COURT----->\n");
    while(ch != 3)
    {
        printf("\t1. Admin \n\t2. Customer \n\t3. Exit \n\tEnter the choice: ");
        scanf("%d",&ch);
        switch(ch)
        {
            case 1 : printf("\n<-----ADMIN PORTAL----->\n\n");
                    printf("\tEnter the admin id: ");
                    scanf("%d",&ad);
                    if(ad == 123)
                    {
                        printf("\n\tLogged in as ADMIN\n");
                        ch=0;        //TO GO BACK TO ADMIN ADMIN
                        while(ch1 != 5)
                        {

                            printf("\n\t1. Add an item \n\t2. Remove an item \n\t3.
Item Menu \n\t4. Total Sale \n\t5. Back \n\tEnter the choice : ");
                            scanf("%d",&ch1);
                            printf("\n");

```



```

switch(ch1)
{
    case 1 : create_Menu();
            break;
    case 2 : delete_Menu();
            break;
    case 3 : display_Menu();
            break;
    case 4 : printf("\tTotal Sale: %d\n",total);
            break;
    case 5 : break;
    default : printf("\tInvalid Response\n");
}
}
break;
case 2 : printf("\n<-----CUSTOMER
PORTAL----->\n\n");
while(ch2 != 5)
{
    printf("\t1. Item Menu \n\t2. Order \n\t3. Checkout \n\t4.
Display Waiting List \n\t5. Back \n\tEnter the choice: ");
    scanf("%d",&ch2);
    printf("\n");
    switch(ch2)
    {
        case 1 : display_Menu();
                break;
        case 2 : if(start1 == NULL)
                    printf("\tMenu is Empty\n");
                else if(table < 5)
                {
                    create_Token(++token);
                }
                else
                {

```

```

        printf("\tYou are in waiting list\n");
        ++waiting;
        insert_Wait(++token);
    }
    break;
case 3 : checkout_Token();
    if(table > 0 && table < 5)
    {
        if(waiting > 0)
        {
            --waiting;
            create_Token(remove_Wait());
        }
    }
    break;
case 4 : if(waiting>0)
    {display_Wait();}
    else{
        printf("WOW book your tables asap");
    }
    break;
case 5 : break;
default : printf("\tInvalid Response\n");
}
if(ch1 == 1)
    break;
}
break;
case 3 : break;
default : printf("\tInvalid Response\n");
}
}
return 0;
}

```

Output ::

```
D:\youtubecprogram\hotelman\bin\Debug\hotelman.exe

***** |MAIN MENU| *****
*****
*Please enter your choice for menu*:

Enter 1 -> Book a room
*****
Enter 2 -> View Customer Record
*****
Enter 3 -> Delete Customer Record
*****
Enter 4 -> Search Customer Record
*****
Enter 5 -> Edit Record
*****
Enter 6 -> Exit
*****
Current date and time : Fri Feb 18 23:02:33 2022
*****
```