Project Name :

Hotel Management System

Description :

This Project Is Made On The Topic Of Hotel Management. We As A Team Worked On This Project. This Is Purely Made By Our Skills Without Any Copy Pasting. We Covered Almost Every Concept We Have Read So Far In Our Course.

About Project :

First Of All , This Program Will Show Up A Menu Containing 4 Choices And Will Ask From User To Choose One From Them. Choices Include Room Booking , Customers Details , History Of Allocated Rooms , Check Room Status , Modification Of Records , Deleting Records , Finding Bill. Comments Has Been Used Almost After Every Line To Show That We Worked Logically.

Code :

```
//C++ PROJECT FOR HOTEL MANAGEMENT SYSTEM MADE BY ZEESHAN , HOONDRAJ AND SYED SHAHZAIB

// LIBRARIES
#include <iostream> // INPUT AND OUTPUT LIBRARY ( Taking Inputs , Showing Outputs )
#include <fstream> // FILE HANDLING LIBRARY ( Writing And Reading In Files )
#include <stdio.h> // STANDARD INPUT OUTPUT LIBRARY ( Getting Input From Hardware )
#include <stdlib.h> // STANDARD LIBRARY ( Memory Allocation )
#include <conio.h> // TO MAKE CONSOLE SCREEN STAY

using namespace std; // SCOPE IDENTIFIERS

//START OF STRUCTURE
struct hotel
{
    int room_no;
    char name[30]; // NAME OF SIZE 30 MAXIMUM
    char address[50]; // ADDRESS OF 50 SIZE
    char phone[11]; // 11 DIGIT PHONE NUMBER

    // FUNCTIONS DECLARATION
    void main_menu();              // TO DISPLAY MAIN MENU
    void add();                       // TO BOOK A ROOM
    void display();         // TO DISPLAY CUSTOMER RECORDS
    void rooms();                  // TO DISPLAY ALLOCATED ROOMS
    void edit();                   // TO EDIT CUSTOMERS RECORD
    int check(int);                // TO CHECK ROOM STATUS
    void modify(int);              // TO MODIFY THE RECORDS
    void delete_rec(int);          // TO DELETE A RECORD
    void bill(int);                  // TO FIND BILL
};
//END OF STRUCTURE

//FOR DISPLAYING MAIN MENU
void hotel::main_menu() // WE USE :: WHEN WE DECLARE FUNCTION OUTSIDE STRUCTURE
{
    int choice = 0;
    while (choice != 5) // CONDITION UNTIL CHOICE OF USER IS 5
    {
        system("cls"); // TO CLEAR SCREEN
        cout << "\n\t\t\t\t*******";
        cout << "\n\t\t\t\t HOTEL MANAGEMENT SYSTEM ";
        cout << "\n\t\t\t\t       MAIN MENU ";
        cout << "\n\t\t\t\t*******";
        cout << "\n\n\t\t\t1.Book A Room";
        cout << "\n\t\t\t2.Customer Records";
        cout << "\n\t\t\t3.Rooms Allotted";
        cout << "\n\t\t\t4.Edit Record";
        cout << "\n\t\t\t5.Exit";
        cout << "\n\n\t\t\tEnter Your Choice: ";
        cin >> choice; // TAKING CHOICE FROM USER

        switch (choice) // CONTROL STRUCTURE
        {
        case 1:      add(); /* FUNCTION CALL */ break;

        case 2: display(); /* FUNCTION CALL */ break;

        case 3: rooms(); /* FUNCTION CALL */ break;
```

```
        case 4:      edit(); /* FUNCTION CALL */ break;

        case 5: break; // BREAK IS USED TO TERMINATE SWITCH CASE

        default: // IF ALL CASES ARE FALSE
        {
            cout << "\n\n\t\t\tERROR !! Please Write A Number Between 1-5";
            cout << "\n\t\t\tPress Any Key To Continue : ) ";
        }
        }
    }
}
//END OF MENU FUNCTION

//FUNCTION FOR BOOKING A ROOM
void hotel::add() // WE USE :: WHEN WE DECLARE FUNCTION OUTSIDE STRUCTURE
{
    system("cls");
    int r, flag; // FLAG IS BOOLEAN VARIABLE WHOSE CONTAIN "TRUE" OR "FALSE"
    ofstream /* TO WRITE IN FILE */ fout("Record.dat", ios::app); // SYNTAX FOR
WRITING IN FILE

    cout << "\n Enter Customer Details";
    cout << "\n ----------------------";
    cout << "\n\n Room No: ";
    cout << "\n Total No. Of Rooms - 50";
    cout << "\n Ordinary Rooms From 1 - 30";
    cout << "\n Luxuary Rooms From 31 - 45";
    cout << "\n Royal Rooms From 46 - 50";
    cout << "\n Enter The Room No. You Want To Stay In :- " << endl;
    cin >> r; // TAKING CHOICE FROM USER

    flag = check(r); // FUNCTION FOR CHECKING

    if (flag)
        cout << "\n Sorry !!! Room Is Already Booked";
    else
    {
        room_no = r;
        cout << "Enter The Information Asked Below : " << endl << endl;
        cout << " Name: ";
        cin >> name;
        cout << " Address: ";
        cin >> address;
        cout << " Phone No: ";
        cin >> phone;

        fout.write((char*)this, sizeof(hotel)); // SYNTAX FOR WRITING INTO FILE
        cout << "\n Congrats !! Room Is Booked...!!!";
    }
    cout << "\n Press Any Key To Continue : ) ";
    fout.close(); // CLOSING FILE
    _getch();
}
//END OF BOOKING FUNCTION

//FUNCTION FOR DISPLAYING A SPECIFIC CUSTOMER`S RECORD
void hotel::display() // WE USE :: WHEN WE DECLARE FUNCTION OUTSIDE STRUCTURE
{
    system("cls");
    ifstream /* READING FROM  FILE*/ fin("Record.dat", ios::in); // SYNTAX FOR READING
FROM FILE
```

```
    int r, flag;

    cout << "\n Enter Room No. To Find Particular Customer`s Details :- " << endl;
    cin >> r; // TAKING ROOM NUMBER FROM USER

    while (!fin.eof()) // CONDITION OF READING FILE TILL END OF FILE
    {
        fin.read((char*)this, sizeof(hotel)); // sizeof IS SIZE OF VARIABLES DECLARED
IN STRUCTURE
        if (room_no == r)
        {
            system("cls");
            cout << "\n Customer Details";
            cout << "\n ----------------";
            cout << "\n\n Room No: " << room_no;
            cout << "\n Name: " << name;
            cout << "\n Address: " << address;
            cout << "\n Phone No: " << phone;
            flag = 1; // TRUE
            break;
        }
    }
    if (flag == 0) // FALSE
        cout << "\n Sorry Room No. Not Found :( ";
    cout << "\n\nPress Any Key To Continue : ) ";
    fin.close(); // FILE CLOSED
    _getch();
}
//END OF DISPLAY FUNCTION

//FUNCTION TO DISPLAY ALL ROOMS OCCUPIED
void hotel::rooms() // WE USE :: WHEN WE DECLARE FUNCTION OUTSIDE STRUCTURE
{
    system("cls");
    ifstream/* READING FROM FILE */ fin("Record.dat", ios::in); // SYNTAX OF READING
FILE
    cout << "\n\t\t\t    List Of Rooms Allotted";
    cout << "\n\t\t\t    ----------------------";
    cout << "\n\n Room No.\tName\t\tAddress\t\t\t\tPhone No.\n";
    while (!fin.eof()) // CONDITION OF READING FILE TILL END OF FILE
    {
        fin.read((char*)this, sizeof(hotel));
        cout << "\n\n " << room_no << "\t\t" << name;
        cout << "\t\t" << address << "\t\t" << phone;
    }
    cout << "\n\n\n\t\t\tPress Any Key To Continue : ) ";
    fin.close(); // FILE CLOSE
    _getch();
}
// FUCNTION ENDED

// FUNCTION FOR EDITING RECORDS AND FOR BILL
void hotel::edit() // WE USE :: WHEN WE DECLARE FUNCTION OUTSIDE STRUCTURE
{
    system("cls");
    int choice, r;
    cout << "\n EDIT MENU";
    cout << "\n ---------";
    cout << "\n\n 1.Modify Customer Record";
    cout << "\n 2.Delete Customer Record";
    cout << "\n 3.Bill Of Customer";
    cout << "\n Enter your choice: ";
```

```
    cin >> choice; // TAKING CHOICE FROM USER
    system("cls");
    cout << "\n Enter Room No: ";
    cin >> r; // TAKING ROOM NO FROM USER

    switch (choice) // CONTROL STRUCTURE
    {
    case 1:   modify(r);/* FUNCTION CALL */break;

    case 2:   delete_rec(r);/* FUNCTION CALL */break;

    case 3: bill(r);/* FUNCTION CALL */break;

    default: cout << "\n ERROR !! Please Write Numbers Between 1-3 ";
    }
    cout << "\n Press Any Key To Continue : ) ";
    _getch();
}
// FUNCTION ENDED

int hotel::check(int r) // CHECKING RECORDS STORED IN FILE
{
    int flag = 0;
    ifstream /* READING FROM FILE */ fin("Record.dat", ios::in); // SYNTAX OF READING
FILE
    while (!fin.eof())
    {
        fin.read((char*)this, sizeof(hotel));
        if (room_no == r)
        {
            flag = 1;break;
        }
    }
    fin.close(); // FILE CLOSED
    return(flag);
}

// FUNCTION TO MODIFY CUSTOMERS RECORD
void hotel::modify(int r) // WE USE :: WHEN WE DECLARE FUNCTION OUTSIDE STRUCTURE
{
    long pos, flag = 0; // LONG IS INTEGER THAT TAKES 4 BYTES
    fstream /* TO READ AND WRITE BOTH IN FILE */file("Record.dat", ios::in | ios::out
| ios::binary); // TO OPEN FILE IN BINARY MODE

    while (!file.eof())
    {
        pos = file.tellg(); // tellg() IS USED TO KNOW WHERE THE GET POINTER IS IN
FILE
        file.read((char*)this, sizeof(hotel));

        if (room_no == r)
        {
            cout << "\n Enter New Details";
            cout << "\n -----------------";
            cout << "\n Name: ";
            cin >> name;
            cout << " Address: ";
            cin >> address;
            cout << " Phone no: ";
            cin >> phone;
```

```
            file.seekg(pos);  // seekg(pos) IS USED TO MOVE GET POINTER TO DESIRED
LOCATION
            file.write((char*)this, sizeof(hotel)); // WRITING IN FILE
            cout << "\n CONGRATS !! Record Is Modified :)";
            flag = 1;break;
        }
    }
    if (flag == 0) // FALSE
        cout << "\n Sorry Room No. Not Found :(";
    file.close(); // FILE CLOSED
    _getch();
}
//END OF MODIFY FUNCTION

//FUNCTION FOR DELETING RECORD
void hotel::delete_rec(int r) // WE USE :: WHEN WE DECLARE FUNCTION OUTSIDE STRUCTURE
{
    int flag = 0;
    char ch;
    ifstream fin("Record.dat", ios::in); // READING FROM FILE
    ofstream fout("temp.dat", ios::out); // WRITING IN TEMPORARY FILE

    while (!fin.eof()) // CONDITION OF READING FILE TILL END OF FILE
    {
        fin.read((char*)this, sizeof(hotel));
        if (room_no == r)
        {
            cout << "\n Name: " << name;
            cout << "\n Address: " << address;
            cout << "\n Phone No: " << phone;
            cout << "\n\n Do You Want To Delete This Record(y/n): ";
            cin >> ch; // TAKE CHOICE FROM USER

            if (ch == 'n')
                fout.write((char*)this, sizeof(hotel)); // RECORD WILL NOT DELETE
            flag = 1; // TRUE
        }
        else
            fout.write((char*)this, sizeof(hotel)); // RECORD WILL NOT DELETE
    }

    fin.close();
    fout.close();

    if (flag == 0) // FALSE
        cout << "\n Sorry Room No. Not Found :(";
    else
    {
        remove("Record.dat"); // DELETE RECORD
        rename("temp.dat", "Record.dat"); // RENAME THE TEMPORARY FILE WITH ORIGINAL
FILE NAME
    }
    _getch();
}
// END OF DELETE FUNCTION

// FUNCTION FOR CUSTOMER`S BILL
void hotel::bill(int r) // WE USE :: WHEN WE DECLARE FUNCTION OUTSIDE STRUCTURE
{
    hotel h1;
    ifstream f1; // READING FILE
    f1.open("record.dat", ios::in | ios::binary);
```

```
    if (!f1)
        cout << "Can Not Open :(";
    else
    {
        f1.read((char*)&h1, sizeof(hotel));
        while (f1)

        {
            f1.read((char*)&h1, sizeof(hotel));
        }

        if (h1.room_no == r)
        {
            if (h1.room_no >= 1 && h1.room_no <= 30)
                cout << "Your Bill Is = 20000 RS. ONLY";

            else if (h1.room_no >= 35 && h1.room_no <= 45)
                cout << "Your Bill Is = 50000 RS. ONLY";

            else
                cout << "Your Bill Is = 70000 RS. ONLY";
        }

        else
        {
            cout << "Room Not Found :(";
        }
    }
    f1.close(); // FILE CLOSED
    _getch();
}
//END OF BILLING FUNCTION

//START OF MAIN PROGARM
int main()
{
    hotel h; // h IS STRUCTURE OBJECT

    system("cls");
    cout << "********************";
    cout << endl << "              Group Project Of CP";
    cout << endl << " Made By : Zeeshan , Hoondraj And Syed Shahzaib" << endl;
    cout << "********************";
    cout << endl << endl << "Press Any Key To Continue :) ";

    h.main_menu();
    return 0;
}
//END OF MAIN PROGRAM
```

Output :

Main Menu

```
                    *******
                     HOTEL MANAGEMENT SYSTEM
                           MAIN MENU
                    *******

              1.Book A Room
              2.Customer Records
              3.Rooms Allotted
              4.Edit Record
              5.Exit

              Enter Your Choice:
```

Booking Room

```
Enter Customer Details
----------------------

Room No:
Total No. Of Rooms - 50
Ordinary Rooms From 1 - 30
Luxuary Rooms From 31 - 45
Royal Rooms From 46 - 50
Enter The Room No. You Want To Stay In :-
```

Giving Informations For Room Booking

```
Enter Customer Details
----------------------

Room No:
Total No. Of Rooms - 50
Ordinary Rooms From 1 - 30
Luxuary Rooms From 31 - 45
Royal Rooms From 46 - 50
Enter The Room No. You Want To Stay In :-
12
Enter The Information Asked Below :

Name: Hoondraj
Address: i10
Phone No: 099999999

Congrats !! Room Is Booked...!!!
Press Any Key To Continue : )
```

Choice # 2

```
Enter Room No. To Find Particular Customer`s Details :-
```

Customer's Detail

```
Customer Details
----------------

Room No: 23
Name: hoondraj
Address: dfgsd
Phone No: 6536546

Press Any Key To Continue : )
```

List Of All Rooms Allocated

```
                    List Of Rooms Allotted
                    ----------------------

Room No.        Name            Address                         Phone No.

28              hitesh          kumar           islmbd
47              HOONDRAJ                I-10            03170839048
23              hoondraj                dfgsd           6536546
48              hoondraj                tyt             9867556
22              FGDGHDH         GFH4RTY         46545634
30              Hitesh          islamabad               0333-7655233▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒p[k¢≈▣
2               hitesh          i10             2222222
12              Hoondraj                i10             099999999
12              Hoondraj                i10             099999999

                    Press Any Key To Continue : )
```

Editing Menu

```
EDIT MENU
---------

1.Modify Customer Record
2.Delete Customer Record
3.Bill Of Customer
Enter your choice:
```

Modifying Details

```
Enter Room No: 23

Enter New Details
-----------------
Name: Zeeshan
Address: pindi
Phone no: 875r78888

CONGRATS !! Record Is Modified :)
```

Deleting Detail

```
Enter Room No: 12

Name: Hoondraj
Address: i10
Phone No: 099999999

Do You Want To Delete This Record(y/n):
```

Printing Bill

```
 Enter Room No: 2
Your Bill Is = 20000 RS. ONLY
```

Choice # 5 Exit

```
                        *******
                         HOTEL MANAGEMENT SYSTEM
                               MAIN MENU
                        *******

                    1.Book A Room
                    2.Customer Records
                    3.Rooms Allotted
                    4.Edit Record
                    5.Exit

                    Enter Your Choice: 5

C:\Users\Hitesh\source\repos\C++ Project 1st semester\x64\Debug\C++ Project 1st semester.exe (process 12412) exited with
 code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the conso
le when debugging stops.
Press any key to close this window . . .
```