C++ PROJECT

\_\_\_\_ HOTEL MANAGEMENT SYSTEM\_\_\_\_



Team Mates:

M Reethu Priya(AP23110011128)

T Naga Sri(AP23110010715)

S. Rupali(AP23110011081)

G. Geethika(AP23110011104)

THE CODE BELOW INCLUDES HOTEL MANAGEMENT SYSTEM

THIS CODE IS USED TO DO THE FOLLOWING :

1 ADDING A GUEST

2 CHECK OUT

3 DISPLAY ROOMS

4 DISPLAY GUESTS

5 EXIT

SOURCE CODE :

#include <iostream>

#include <string>

#include <vector>

#include <set>

using namespace std;

class Guest {

public:

string name;

string contact;

int roomNumber;

Guest(string guestName, string guestContact, int assignedRoom)

: name(guestName), contact(guestContact), roomNumber(assignedRoom) {}

void displayGuestInfo() {

cout << "Guest Name: " << name << ", Contact: " << contact

<< ", Room Number: " << roomNumber << endl;

}

};

class HotelManagementSystem {

vector<Guest> guests;

set<int> occupiedRooms;

public:

void addGuest(string name, string contact) {

cout << "Available Rooms:\n";

for (int floor = 1; floor <= 5; ++floor) {

for (int room = 1; room <= 10; ++room) {

int roomNumber = floor \* 100 + room;

if (occupiedRooms.find(roomNumber) == occupiedRooms.end()) {

cout << "Room Number: " << roomNumber << ", Available: Yes\n";

}

}

}

int roomNumber;

cout << "Enter room number to assign: ";

cin >> roomNumber;

if (occupiedRooms.find(roomNumber) != occupiedRooms.end()) {

cout << "Room " << roomNumber << " is already occupied.\n";

return;

}

if (!((roomNumber >= 101 && roomNumber <= 110) ||

(roomNumber >= 201 && roomNumber <= 210) ||

(roomNumber >= 301 && roomNumber <= 310) ||

(roomNumber >= 401 && roomNumber <= 410) ||

(roomNumber >= 501 && roomNumber <= 510))) {

cout << "Invalid room number.\n";

return;

}

guests.push\_back(Guest(name, contact, roomNumber));

occupiedRooms.insert(roomNumber);

cout << "Room assigned to guest successfully.\n";

}

void checkOut(int roomNumber) {

if (!((roomNumber >= 101 && roomNumber <= 110) ||

(roomNumber >= 201 && roomNumber <= 210) ||

(roomNumber >= 301 && roomNumber <= 310) ||

(roomNumber >= 401 && roomNumber <= 410) ||

(roomNumber >= 501 && roomNumber <= 510))) {

cout << "Invalid room number.\n";

return;

}

if (occupiedRooms.find(roomNumber) == occupiedRooms.end()) {

cout << "Room " << roomNumber << " is already available.\n";

return;

}

for (auto it = guests.begin(); it != guests.end(); ++it) {

if (it->roomNumber == roomNumber) {

guests.erase(it);

break;

}

}

occupiedRooms.erase(roomNumber);

cout << "Room " << roomNumber << " is now available.\n";

}

void displayRooms() {

cout << "\nRoom List:\n";

for (int floor = 1; floor <= 5; ++floor) {

for (int room = 1; room <= 10; ++room) {

int roomNumber = floor \* 100 + room;

cout << "Room Number: " << roomNumber << ", Available: "

<< (occupiedRooms.find(roomNumber) == occupiedRooms.end() ? "Yes" : "No") << endl;

}

}

}

void displayGuests() {

if (guests.empty()) {

cout << "\nNo guests currently staying at the hotel.\n";

return;

}

cout << "\nGuest List:\n";

for (auto &guest : guests) {

guest.displayGuestInfo();

}

}

};

int main() {

HotelManagementSystem hotel;

int choice;

while (true) {

cout << "\n--- Hotel Management System ---\n";

cout << "1. Add Guest\n";

cout << "2. Check Out\n";

cout << "3. Display Rooms\n";

cout << "4. Display Guests\n";

cout << "5. Exit\n";

cout << "Enter your choice: ";

cin >> choice;

switch (choice) {

case 1: {

string name, contact;

cin.ignore();

cout << "Enter guest name: ";

getline(cin, name);

cout << "Enter guest contact: ";

getline(cin, contact);

hotel.addGuest(name, contact);

break;

}

case 2: {

int roomNumber;

cout << "Enter room number to check out: ";

cin >> roomNumber;

hotel.checkOut(roomNumber);

break;

}

case 3:

hotel.displayRooms();

break;

case 4:

hotel.displayGuests();

break;

case 5:

cout << "Exiting the system. Goodbye!\n";

return 0;

default:

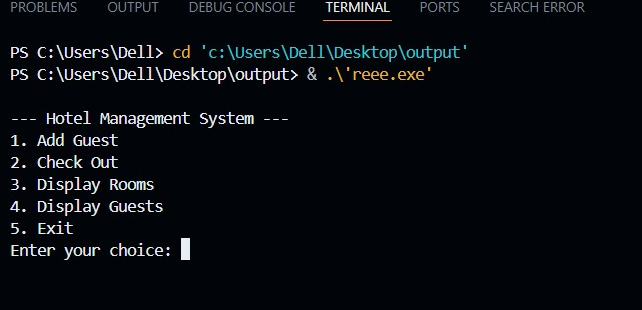
cout << "Invalid choice. Please try again.\n";

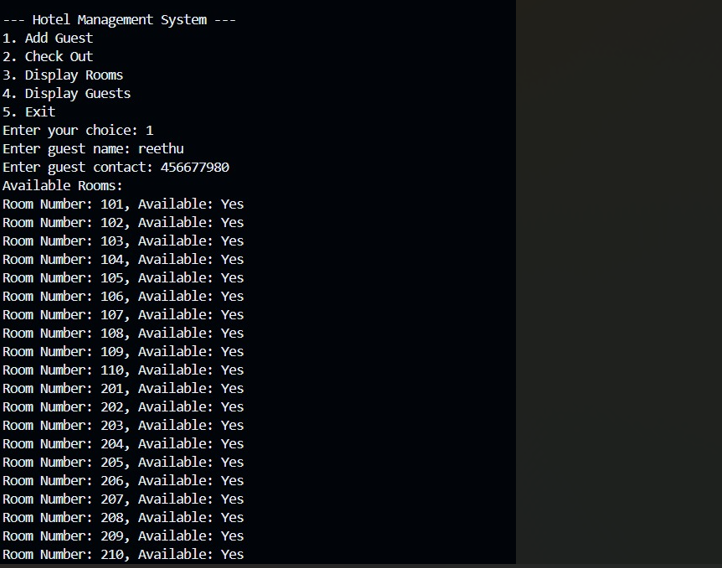
}

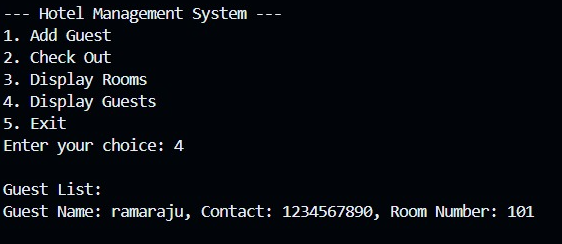
}

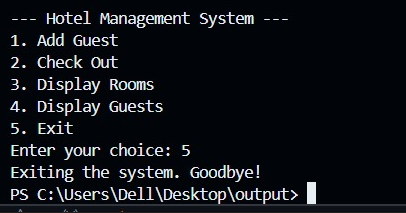
}

OUTPUT :









LINK FOR OUTPUT

<https://docs.google.com/document/d/1JQ5umE-3PPfM-dsBVZdxRpCj3cGZZVrt_hhXwlJ2-zI/edit?usp=drivesdk>

**Functions used in the Code**

**1. Class Guest Functions**

* displayGuestInfo():  
  Displays guest details (name, contact, and room number).

**2. Class HotelManagementSystem Functions**

* addGuest(string name, string contact):  
  Adds a new guest and assigns a room.
* checkOut(int roomNumber):  
  Removes a guest and marks their room as available.
* displayRooms():  
  Displays the availability of all rooms in the hotel.
* displayGuests():  
  Displays the details of all current guests in the hotel.

**3. Main Program Utility**

* **Menu and Input Handling in main()**:  
  Manages the user interface, processes input, and calls the appropriate functions.