//SPDX-License-Identifier: MIT

pragma solidity ^0.8.14;

contract hotelBooking{

struct hotelRoom{

string categoryName;

uint tariff;

bool occupied;

uint review;

uint reviewNo;

address costumerBooked;

bool booked;

}

struct costumer{

string name;

string addre;

uint custIdNo;

}

uint private custCount;

uint private noOfRooms;

address public owner;

modifier onlyowner{

require(msg.sender == owner , "Only owner has the rights to add rooms");

\_;

}

mapping(uint=>hotelRoom) public hotelRoomDetails;

mapping(address=>costumer) public costumerDetails;

constructor(){

owner = msg.sender;

setHotelRoom(1, "Royal", 10);

setHotelRoom(2, "Premium", 5);

setHotelRoom(3, "Delux", 3);

}

function setHotelRoom(uint \_roomNo , string memory \_categoryName, uint \_tariff) public onlyowner{

hotelRoomDetails[\_roomNo].categoryName = \_categoryName;

hotelRoomDetails[\_roomNo].tariff = \_tariff;

hotelRoomDetails[\_roomNo].review = 0;

hotelRoomDetails[\_roomNo].reviewNo = 0;

noOfRooms++;

}

function setcust(address \_addr , string memory \_name ,string memory \_addre) public {

costumerDetails[\_addr].name = \_name;

costumerDetails[\_addr].addre = \_addre;

costumerDetails[\_addr].custIdNo = custCount;

custCount++ ;

}

function payToBook() public payable {

if(msg.value==10){

bookRoomRoyal();

}

else if(msg.value==5){

bookRoomPremium();

}

else if(msg.value==3){

bookRoomDeluxe();

}

else{

revert();

}

}

function bookRoomRoyal() internal {

require(hotelRoomDetails[1].booked==false , "Room is already booked");

hotelRoomDetails[1].booked = true;

hotelRoomDetails[1].costumerBooked = msg.sender;

}

function bookRoomPremium() internal {

require(hotelRoomDetails[2].booked==false , "Room is already booked");

hotelRoomDetails[2].booked = true;

hotelRoomDetails[2].costumerBooked = msg.sender;

}

function bookRoomDeluxe() internal {

require(hotelRoomDetails[3].booked==false , "Room is already booked");

hotelRoomDetails[3].booked = true;

hotelRoomDetails[3].costumerBooked = msg.sender;

}

function checkInRoyal() external {

require(hotelRoomDetails[1].costumerBooked==msg.sender , "This room has not been booked by you.");

hotelRoomDetails[1].occupied = true;

}

function checkInPremium() external {

require(hotelRoomDetails[2].costumerBooked==msg.sender , "This room has not been booked by you.");

hotelRoomDetails[2].occupied = true;

}

function checkInDeluxe() external {

require(hotelRoomDetails[3].costumerBooked==msg.sender , "This room has not been booked by you.");

hotelRoomDetails[3].occupied = true;

}

function checkOutRoyal(uint \_rating) external {

require(hotelRoomDetails[1].costumerBooked==msg.sender , "This room has not been booked by you.");

hotelRoomDetails[1].occupied = false;

hotelRoomDetails[1].booked = false;

hotelRoomDetails[1].review = (\_rating+ (hotelRoomDetails[1].review\*hotelRoomDetails[1].reviewNo))/(hotelRoomDetails[1].reviewNo+1);

hotelRoomDetails[1].reviewNo++;

}

function checkOutPremium(uint \_rating) external {

require(hotelRoomDetails[2].costumerBooked==msg.sender , "This room has not been booked by you.");

hotelRoomDetails[2].occupied = false;

hotelRoomDetails[2].booked = false;

hotelRoomDetails[2].review = (\_rating+ (hotelRoomDetails[2].review\*hotelRoomDetails[2].reviewNo))/(hotelRoomDetails[2].reviewNo+1);

hotelRoomDetails[2].reviewNo++;

}

function checkOutDeluxe(uint \_rating) external {

require(hotelRoomDetails[3].costumerBooked==msg.sender , "This room has not been booked by you.");

hotelRoomDetails[3].occupied = false;

hotelRoomDetails[3].booked = false;

hotelRoomDetails[3].review = (\_rating+ (hotelRoomDetails[3].review\*hotelRoomDetails[3].reviewNo))/(hotelRoomDetails[3].reviewNo+1);

hotelRoomDetails[3].reviewNo++;

}

}