KENDRIYA VIDYALAYA,

# ORDNANCE FACTORY, MEDAK



ACADEMIC YEAR 2021-2022

PROJECT REPORT ON HOTEL MANAGEMENT

SYSTEM

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| NAME |  | : | GORILE SHAILAJA |
| CLASS |  | : | XII - A |
| SUBJECT |  | : | COMPUTER SCIENCE |
| SUBJECT CODE |  | : | 083 |
| PROJECT GUIDE |  | : | Ms. NISHA SHARMA (PGT) |

KENDRIYA VIDYALAYA,

ORDNANCE FACTORY, MEDAK

# KENDRIYA VIDYALAYA, ORDNANCE FACTORY, MEDAK



## CERTIFICATE

This is to certify that GORILE SHAILAJA has successfully completed the project work entitled HOTEL

MANAGEMENT SYSTEM in the subject COMPUTER

SCIENCE ( 083 ) laid down in the regulations of CBSE for the purpose of Practical Examination in Class XII to be held in

KENDRIYA VIDHYALAYA, ORDNANCE FACTORY, MEDAK.

External Examiner : Internal Examiner :

Name : \_\_\_\_\_\_\_\_\_\_\_ Ms. NISHA SHARMA ( PGT CS )

Signature : TEACHER IC

Principal : \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## ACKNOWLEDGEMENT

First of all, I heartly thank my computer science teacher, MISS NISHA SHARMA for her incomparable efforts, support and constant co-operation indeed towards us in the Completion of this project. She has also been the vital source of encouragement for me through the working of this project. Finally I would also grateful to my friends and parents who devoted their auspicious time in completion of this project.

Thankyou

### Gorile shailaja

|  |  |  |
| --- | --- | --- |
|  | INDEX |  |
| S.No | Topic | Page no. |
| 01 | Certificate | II |
| 02 | Acknowledgement | III |
| 03 | Introduction | 1 |
| 04 | Hardware and Software Requirements | 4 |
| 05 | Operating Instructions | 5 |
| 06 | Source Code | 7 |
| 07 | Output Screenshots | 40 |
| 08 | Limitations | 44 |
| 09 | Future Scope | 45 |
| 10 | Bibliography | 45 |

# HOTEL MANAGEMENT SYSTEM

INTRODUCTION

This Project on Hotel Management is a general software developed (using Python) to simplify hotel operations by automating them. In this project, “AnCasa” is the project’s hotel name. It covers major aspects of hotel management; it can perform the following operations- Hotel Booking, Provide you with Hotel Rooms Info, Room Service, Billing and Record-Keeping.

Functionalities Provided By Python Project On HOTEL MANAGEMENT SYSTEM Are As Follows:

It can perform the following operations- Hotel Booking, Provide you with Hotel Rooms Info, Room Service, Billing and Record-Keeping.

Input Data And Validation Of Python Project On Parking

System

1. All the fields are validated and does not take invalid values.
2. Each form cannot accept blank value fields.

FUNCTIONS CREATED

Home()- Function to display the project’s main screen i.e. the home page of the project or you can say the main menu for selecting the desired operation to perform.

Date(str)- Function to validate date entered by the user/customer.

Booking()- Function for booking room in hotel by entering user/customer details.

Room\_Info()- Function to provide users/customers with hotel rooms information(i.e. about room amenities).

Restaurant()- Function for room service which provides user/customer with the restaurant’s menu card to order food at the room.

Payment()- Function for payment of hotel room and restaurant bill generation at the time of check-out.

Record()- Function for keeping records of customers stayed in the hotel.

HARDWARE AND SOFTWARE

## REQUIREMENTS

### MINIMUM HARDWARE REQUIREMENTS

1. OPERATING SYSTEM :
   1. Windows ® 7 or Above
   2. MAC OS X 10.11 or Higher, 64-Bit
   3. LINUX : RHEL 6/7, 64-Bit

( Almost all libraries also work in Ubuntu )

1. PROCESSORS : Intel Atom ® or Intel ® Core ™ i3

Processor

1. RAM : 4 GB RAM

1. HARD DISK : Min. 5 GB of free disk space

1. PRINTER : Required if print is needed [ HARD COPY ]

SOFTWARE REQUIREMENTS :

1. Windows OS
2. Python 3.0 or Above

OPERATING INSTRUCTIONS

1. Open This PC
2. Go to DVD RW Drive
3. Select HOTEL MANAGEMENT SYSTEM MAIN CODE (.py)
4. Open with Python IDLE
5. Run the code by pressing F5 key
6. Exit

IV

# SOURCE CODE

DATABASE CODE :

-- phpMyAdmin SQL Dump

-- version 4.7.4

-- https://www.phpmyadmin.net/

--

-- Host: 127.0.0.1:3306

-- Generation Time: Dec 29, 2020 at 05:27 PM

-- Server version: 5.7.19

-- PHP Version: 7.1.9

SET SQL\_MODE = "NO\_AUTO\_VALUE\_ON\_ZERO";

SET AUTOCOMMIT = 0; START TRANSACTION;

SET time\_zone = "+00:00";

/\*!40101 SET

@OLD\_CHARACTER\_SET\_CLIENT=@@CHARACTER\_SET\_C

LIENT \*/;

/\*!40101 SET

@OLD\_CHARACTER\_SET\_RESULTS=@@CHARACTER\_SET\_

RESULTS \*/;

/\*!40101 SET

@OLD\_COLLATION\_CONNECTION=@@COLLATION\_CONN

ECTION \*/;

/\*!40101 SET NAMES utf8mb4 \*/;

--

-- Database: `hotel`

--

DROP DATABASE IF EXISTS `hotel`;

CREATE DATABASE IF NOT EXISTS `hotel`;

USE `hotel`;

-- --------------------------------------------------------

--

-- Table structure for table `bill`

--

DROP TABLE IF EXISTS `bill`;

CREATE TABLE IF NOT EXISTS `bill` (

`bill\_id` bigint(20) NOT NULL AUTO\_INCREMENT,

`book\_id` bigint(20) DEFAULT NULL,

`amount` float(10,2) DEFAULT NULL,

`bill\_date` date DEFAULT NULL,

`gst` int(10) DEFAULT NULL,

`st` int(10) DEFAULT NULL,

PRIMARY KEY (`bill\_id`)

) ENGINE=MyISAM AUTO\_INCREMENT=10 DEFAULT

CHARSET=latin1;

--

-- Dumping data for table `bill`

--

INSERT INTO `bill` (`bill\_id`, `book\_id`, `amount`, `bill\_date`, `gst`,

`st`) VALUES

(1, 1, 770500.00, '2020-12-25', 0, 0),

(2, 2, 500.00, '2020-12-25', 0, 0),

(3, 3, 1600.00, '2020-12-25', 0, 0),

(4, 4, 63000.00, '2020-12-25', 0, 0),

(5, 5, 10500.00, '2020-12-25', 0, 0),

(6, 6, 8000.00, '2020-12-25', 0, 0),

(7, 7, 10500.00, '2020-12-25', 23, 5), (8, 8, 11500.00, '2020-12-25', 23, 5),

(9, 10, -3400.00, '2020-12-29', 23, 5);

-- --------------------------------------------------------

--

-- Table structure for table `booking`

--

DROP TABLE IF EXISTS `booking`;

CREATE TABLE IF NOT EXISTS `booking` (

`book\_id` bigint(20) NOT NULL AUTO\_INCREMENT,

`room\_id` bigint(20) DEFAULT NULL,

`cust\_id` bigint(20) DEFAULT NULL,

`doo` date DEFAULT NULL,

`dol` date DEFAULT NULL,

`advance` float(10,2) DEFAULT NULL,

PRIMARY KEY (`book\_id`)

) ENGINE=MyISAM AUTO\_INCREMENT=11 DEFAULT CHARSET=latin1;

--

-- Dumping data for table `booking`

--

INSERT INTO `booking` (`book\_id`, `room\_id`, `cust\_id`, `doo`,

`dol`, `advance`) VALUES

(1, 1, 1, '2020-02-20', '2020-12-25', 2000.00), (2, 2, 1, '2020-12-23', '2020-12-25', 4500.00), (3, 4, 1, '2020-12-23', '2020-12-25', 3400.00), (4, 1, 1, '2020-11-29', '2020-12-25', 2000.00), (5, 1, 1, '2020-12-20', '2020-12-25', 2000.00), (6, 1, 1, '2020-12-20', '2020-12-25', 4500.00), (7, 1, 1, '2020-12-19', '2020-12-25', 4500.00),

(8, 1, 1, '2020-12-20', '2020-12-25', 4500.00),

(9, 1, 1, '2020-12-20', NULL, 2500.00),

(10, 2, 2, '2020-12-29', '2020-12-29', 3400.00);

-- --------------------------------------------------------

--

-- Table structure for table `customer`

--

DROP TABLE IF EXISTS `customer`;

CREATE TABLE IF NOT EXISTS `customer` (

`id` bigint(20) NOT NULL AUTO\_INCREMENT,

`name` char(50) DEFAULT NULL,

`address` char(100) DEFAULT NULL,

`phone` char(15) DEFAULT NULL,

`email` char(80) DEFAULT NULL,

`id\_proof` char(20) DEFAULT NULL,

`id\_proof\_no` char(25) DEFAULT NULL,

`males` int(2) DEFAULT NULL,

`females` int(2) DEFAULT NULL,

`children` int(2) DEFAULT NULL,

PRIMARY KEY (`id`)

) ENGINE=MyISAM AUTO\_INCREMENT=3 DEFAULT

CHARSET=latin1;

--

-- Dumping data for table `customer`

--

INSERT INTO `customer` (`id`, `name`, `address`, `phone`, `email`,

`id\_proof`, `id\_proof\_no`, `males`, `females`, `children`) VALUES

(1, 'rakesh kumar', 'CF-4 BRIJ VIHAR', '98718168101',

'RAKESH@GMAIL.COM', 'AADHAR CARD', '4544-5656-5656', 1,

1, 2),

(2, 'ajmal khan', 'F-234 BRIJ VIHAR', '456465456',

'AJMAL@GMAIL.COM', 'AADHAR', '4353-4564-5675', 2, 0, 0);

-- --------------------------------------------------------

--

-- Table structure for table `rooms`

--

DROP TABLE IF EXISTS `rooms`;

CREATE TABLE IF NOT EXISTS `rooms` (

`id` int(10) NOT NULL AUTO\_INCREMENT,

`room\_no` int(4) DEFAULT NULL,

`room\_type` char(20) DEFAULT NULL,

`room\_rent` float(10,2) DEFAULT NULL,

`room\_bed` char(20) DEFAULT NULL,

`status` char(20) DEFAULT NULL,

PRIMARY KEY (`id`)

) ENGINE=MyISAM AUTO\_INCREMENT=27 DEFAULT

CHARSET=latin1;

--

-- Dumping data for table `rooms`

--

INSERT INTO `rooms` (`id`, `room\_no`, `room\_type`, `room\_rent`,

`room\_bed`, `status`) VALUES

(1, 1, 'AC', 2500.00, 'Single Bed', 'occupied'),

(2, 2, 'AC', 2500.00, 'Single Bed', 'free'), (3, 3, 'AC', 2500.00, 'Single Bed', 'free'), (4, 4, 'AC', 2500.00, 'Single Bed', 'free'),

(6, 5, 'AC', 3500.00, 'Double Bed', 'free'), (7, 6, 'AC', 3500.00, 'Double Bed', 'free'), (8, 7, 'AC', 3500.00, 'Double Bed', 'free'),

(9, 8, 'Delux', 4500.00, 'Double Bed', 'free'),

(10, 9, 'Delux', 4500.00, 'Double Bed', 'free'),

(11, 10, 'Delux', 4500.00, 'Double Bed', 'free'),

(12, 10, 'Super Delux', 5500.00, 'Double Bed', 'free'), (13, 11, 'Super Delux', 5500.00, 'Double Bed', 'free'),

(14, 11, 'Queen Delight', 6500.00, 'Double Bed', 'free'),

(15, 12, 'Queen Delight', 6500.00, 'Double Bed', 'free'),

(16, 13, 'King Special', 7500.00, 'Double Bed', 'free'), (17, 14, 'King Special', 7500.00, 'Double Bed', 'free'),

(18, 15, 'Super Rich Special', 9500.00, 'Double Bed', 'free'),

(19, 16, 'Super Rich Special', 8500.00, 'Single Bed', 'free'),

(20, 17, 'Delux', 4000.00, 'Single Bed', 'free'),

(21, 18, 'Super Delux', 4500.00, 'Single Bed', 'free'),

(22, 19, 'Super Delux', 4500.00, 'Single Bed', 'free'),

(23, 20, 'AC', 2650.00, 'Single Bed', 'free'),

(24, 23, 'Non-AC', 1500.00, 'Single Bed', 'free'),

(25, 24, 'Non-AC', 1500.00, 'Double Bed', 'free'),

(26, 25, ' AC', 3500.00, 'SINGLE', 'free');

-- --------------------------------------------------------

--

-- Table structure for table `setting`

--

DROP TABLE IF EXISTS `setting`;

CREATE TABLE IF NOT EXISTS `setting` (

`id` int(10) NOT NULL AUTO\_INCREMENT,

`field\_name` char(30) DEFAULT NULL,

`value` char(100) DEFAULT NULL,

PRIMARY KEY (`id`)

) ENGINE=MyISAM AUTO\_INCREMENT=8 DEFAULT CHARSET=latin1;

--

-- Dumping data for table `setting`

--

INSERT INTO `setting` (`id`, `field\_name`, `value`) VALUES

(1, 'hotel\_name', 'Hotel Leela'),

(2, 'address', 'SurajMal Vihar, Delhi -92'),

(7, 'phone', '011-4353534,4654545,456567556'),

(4, 'email', 'leela\_delhi@gmail.com'),

(5, 'gst', '23'),

(6, 'st', '8');

-- --------------------------------------------------------

--

-- Table structure for table `temp`

--

DROP TABLE IF EXISTS `temp`;

CREATE TABLE IF NOT EXISTS `temp` (

`book\_id` bigint(20) NOT NULL DEFAULT '0',

`doo` date DEFAULT NULL,

`dol` date DEFAULT NULL,

`advance` float(10,2) DEFAULT NULL,

`name` char(50) DEFAULT NULL,

`address` char(100) DEFAULT NULL,

`phone` char(15) DEFAULT NULL,

`email` char(80) DEFAULT NULL, `room\_no` int(4) DEFAULT NULL

) ENGINE=MyISAM DEFAULT CHARSET=latin1;

--

-- Dumping data for table `temp`

--

INSERT INTO `temp` (`book\_id`, `doo`, `dol`, `advance`, `name`,

`address`, `phone`, `email`, `room\_no`) VALUES

(1, '2020-02-20', '2020-12-25', 2000.00, 'rakesh kumar', 'CF-4 BRIJ VIHAR', '98718168101', 'RAKESH@GMAIL.COM', 1),

(2, '2020-12-23', '2020-12-25', 4500.00, 'rakesh kumar', 'CF-4 BRIJ VIHAR', '98718168101', 'RAKESH@GMAIL.COM', 2),

(3, '2020-12-23', '2020-12-25', 3400.00, 'rakesh kumar', 'CF-4 BRIJ

VIHAR', '98718168101', 'RAKESH@GMAIL.COM', 4),

(4, '2020-11-29', '2020-12-25', 2000.00, 'rakesh kumar', 'CF-4 BRIJ VIHAR', '98718168101', 'RAKESH@GMAIL.COM', 1),

(5, '2020-12-20', '2020-12-25', 2000.00, 'rakesh kumar', 'CF-4 BRIJ VIHAR', '98718168101', 'RAKESH@GMAIL.COM', 1),

(6, '2020-12-20', '2020-12-25', 4500.00, 'rakesh kumar', 'CF-4 BRIJ VIHAR', '98718168101', 'RAKESH@GMAIL.COM', 1),

(7, '2020-12-19', '2020-12-25', 4500.00, 'rakesh kumar', 'CF-4 BRIJ VIHAR', '98718168101', 'RAKESH@GMAIL.COM', 1),

(8, '2020-12-20', '2020-12-25', 4500.00, 'rakesh kumar', 'CF-4 BRIJ VIHAR', '98718168101', 'RAKESH@GMAIL.COM', 1);

COMMIT;

/\*!40101 SET

CHARACTER\_SET\_CLIENT=@OLD\_CHARACTER\_SET\_CLIE

NT \*/;

/\*!40101 SET

CHARACTER\_SET\_RESULTS=@OLD\_CHARACTER\_SET\_RES

ULTS \*/;

/\*!40101 SET

COLLATION\_CONNECTION=@OLD\_COLLATION\_CONNECTI

ON \*/;

MAIN CODE :

# Project Name : Hotel Management System

# Made by : Kavya

# Shilaja # Pallavi

# session : 2021 - 22

# roll no : roll no

import mysql.connector from datetime import date import time

global conn, cursor

conn = mysql.connector.connect(host='localhost', database='hotel', user='root', password='srujan@12a') cursor = conn.cursor()

#global variables hotel\_name ='' addr ='' phone='' email ='' gst=0 st =0

def made\_by():

clear() msg = '''

Hotel Management System made by : Kavya | Shailaja | Pallavi

Roll No : 1 | 2 | 3

School Name : Kendriya Vidyalaya ODF

Medak

Session : 2021-22

Thanks for evaluating my Project.

\n\n\n

'''

for x in msg:

print(x, end='') time.sleep(0.002)

wait = input('Press any key to Exit.....')

def settings():

global hotel\_name global addr global phone global email global gst global st sql = "select \* from setting;" cursor.execute(sql) records = cursor.fetchall() for record in records: if record[1]=='hotel\_name': hotel\_name = record[2] elif record[1] == 'address':

addr = record[2] elif record[1] == 'phone': phone = record[2] elif record[1] == 'email': email = record[2] elif record[1] == 'gst': gst = record[2] elif record[1] == 'st': st = record[2]

def system\_settings():

clear()

print(' Change System Settings ') print('\*'\*120) print('1. Hotel Name') print('2. Hotel Address') print('3. Phone Number(s)') print('4. Email ID') print('5. Current GST Rate') print('6. Current Service Rate') choice = int(input('Enter your choice :')) field\_name ='' if choice ==1:

field\_name='hotel\_name' elif choice ==2:

field\_name='address' elif choice ==3:

field\_name='phone' elif choice ==4:

field\_name='email' elif choice ==5: field\_name='gst' elif choice ==6: field\_name='st'

value = input('Enter new value :')

sql ='update setting set value= '+value+' where field\_name = "'+ field\_name+'";'

cursor.execute(sql)

wait = input('\n\n\n Record Updated .............Press any key to continue......')

def clear(): for \_ in range(65):

print()

def room\_exist(room\_no):

sql ="select \* from rooms where room\_no ="+room\_no+";" cursor.execute(sql) record = cursor.fetchone() return record

def customer\_exist(cust\_no):

sql = "select \* from customer where id ="+cust\_no+";" cursor.execute(sql) record = cursor.fetchone() return record

def add\_room():

clear()

print('Add New Room - Screen') print('-'\*120)

room\_no = input('\n Enter Room No :')

room\_type = input('\n Enter Room Type( AC/DELUX/Super Delux/Queen Delight/ Kings Special/Super Rich Special) :') room\_rent = input('\n Enter Room Rent (INR) :')

room\_bed = input('\n Enter Room Bed Type(Single/Double/Triple)

:')

sql = 'insert into

rooms(room\_no,room\_type,room\_rent,room\_bed,status) values \

('+room\_no+',"'+ room\_type.upper()+'",'+room\_rent+',"'+room\_bed.upper()+'","free");'

result = room\_exist(room\_no) if result is None:

cursor.execute(sql) else:

print('\n\n\nRoom No ',room\_no, ' already exists in our database')

wait = input('\n\n\n Press any key to continue....')

def modify\_room():

clear()

print(' Change Room Information ')

print('\*'\*120) print('1. Room Type') print('2. Room Rent') print('3. Room Bed')

choice = int(input('Enter your choice :')) field\_name = '' if choice == 1:

field\_name = 'room\_type' elif choice == 2:

field\_name = 'room\_rent' elif choice == 3:

field\_name = 'room\_bed' room\_no = input('Enter room No :') value = input('Enter new value :')

sql = 'update rooms set {} = "{}" where room\_no =

{};'.format(field\_name, value, room\_no) cursor.execute(sql)

wait = input('\n\n\n Record Updated .............Press any key to continue......')

def add\_customer():

clear()

print('Add New Customer - Screen') print('-'\*120)

name = input('\n Enter Customer Name :')

address = input('\n Enter Customer Address:') phone = input('\n Enter Customer Phone NO :') email = input('\n Enter Customer Email ID :') id\_proof = input('\n Enter Customer ID(Aadhar/Passport/DL/VoterID) :')

id\_proof\_no = input('\n Enter Customer ID proof NO :') males = input('\n Enter Total Males :') females = input('\n Enter Total Females :') children = input('\n Enter Total Childeren :')

sql = 'insert into

customer(name,address,phone,email,id\_proof,id\_proof\_no,males,fem ales,children) values \

("{}", "{}", "{}", "{}", "{}", "{}", "{}", "{}", "{}");'.format \

(name, address.upper(), phone, email.upper(), id\_proof.upper(), id\_proof\_no.upper(), males, females, children)

cursor.execute(sql)

print('\n\n\nCustomer Added success fully ...............')

wait = input('\n\n\n Press any key to continue....')

def modify\_customer():

clear()

print(' Change Customer Information ')

print('\*'\*120) print('1. Name') print('2. Address') print('3. Phone No') print('4. Email ID') print('5. ID Proof') print('6. ID Proof No') print('7. Males') print('8. Females') print('9. Childeren') choice = int(input('Enter your choice :')) field\_name = '' if choice == 1:

field\_name = 'name' elif choice == 2:

field\_name = 'address' elif choice == 3:

field\_name = 'phone' elif choice == 4:

field\_name = 'email' elif choice == 5:

field\_name = 'id\_proof' elif choice == 6:

field\_name = 'id\_proof\_no' elif choice == 7:

field\_name = 'males' elif choice == 8:

field\_name = 'females' elif choice == 9:

field\_name = 'children' cust\_no = input('Enter Customer No :') value = input('Enter new value :')

sql = 'update customer set {} = "{}" where id = {};'.format(field\_name, value, cust\_no) cursor.execute(sql) wait = input(

'\n\n\n Record Updated .............Press any key to continue......')

def room\_booking():

room\_id =input('Enter room no to book :') cust\_id = input('Enter customer ID :')

date\_of\_occ = input('Enter date of occupancy (yyyy-mm-dd) :') advance = input('Enter advance amount :')

sql1 = 'update rooms set status = "occupied" where id =

{};'.format(room\_id)

sql2 = 'insert into booking(room\_id,cust\_id,doo,advance) values ("{}", "{}", "{}", "{}");'.format(room\_id, cust\_id, date\_of\_occ, advance)

#print(sql2) #print(sql1)

result = room\_exist(room\_id)

result1 = customer\_exist(cust\_id) if result[5]=='free' and result1 is not None:

cursor.execute(sql1) cursor.execute(sql2)

print('\n\n\nRoom no ', room\_id, 'booked for', cust\_id) elif result[5] !='free':

print('\n Room is not available for booking. Right now it is

:',result[5]) elif result1 is None:

print('Customer does not exist....Please add customer first in our database')

wait = input('\n\n\n Press any key to continue....')

def bill\_generation():

global gst global st

room\_id = input('Enter room no to book :') cust\_id = input('Enter customer ID :')

sql = 'select \* from booking where cust\_id = "{}" and room\_id =

"{}" and dol is NULL;'.format(cust\_id, room\_id) cursor.execute(sql) record = cursor.fetchone() clear()

print('Bill Generation ')

print('-'\*100)

print(' Rooms occupied :',room\_id) dol = date.today() book\_id = record[0] # book\_id doo = record[3] advance = record[5] total\_days = (dol-doo).days result = room\_exist(room\_id) rent = result[3] amount = total\_days\*rent gst\_amount = amount\*int(gst)/100 st\_amount = amount\*int(st)/100 payable\_amount = total\_days\*rent - advance + gst\_amount+st\_amount

print('Date of Occupancy :',doo, '\nDate of Leaving :',dol) print('Total Payable Days : ', total\_days) print('Room Rent Per Day : ', rent) print('Total Amount :',amount) print('Advance :',advance,'\nGST ({}) : {}

'.format(gst,gst\_amount), '\nService Tax ({}) :

{}'.format(st,st\_amount))

print('Amount Payable :',payable\_amount)

sql1 = 'update rooms set status ="free" where room\_no =

"{}";'.format(room\_id)

sql2 = 'update booking set dol = "{}" where room\_id = "{}" and cust\_id = "{}";'.format(str(dol), room\_id, cust\_id)

sql3 = 'insert into bill(book\_id,amount,bill\_date,gst,st) values("{}", "{}", "{}", "{}", "{}");'.format(str(book\_id,), str(payable\_amount), str(dol), str(gst), str(st)) cursor.execute(sql1) cursor.execute(sql2) cursor.execute(sql3)

wait = input('\n\n\n Press any key to continue....')

def search\_rooms():

room\_no = input('Enter Room No :')

sql ='select \* from rooms where room\_no ='+room\_no +';' cursor.execute(sql) record = cursor.fetchone() clear()

print('Room Status') print('\*'\*120)

print('Room NO :',record[1]) print('Room Rent :',record[2]) print('Room Bed :',record[3]) print('Room Status :',record[4])

wait = input('\n\n\nPress any key to continue......')

def search\_customer():

clear()

print('Search Customer DataBase') print('\*'\*120) print('1. Customer Name') print('2. Customer Address') print('3. Customer Phone') print('4. Customer Email') print('5. Address Proof') print('6. Address Proof ID') choice = int(input('Enter your choice : ')) field\_name ='' if choice ==1:

field\_name = 'name' elif choice ==2:

field\_name = 'address' elif choice ==3:

field\_name = 'phone' elif choice ==4:

field\_name = 'email' elif choice ==5:

field\_name = 'id\_proof' elif choice ==6:

field\_name = 'id\_proof\_no'

value = input('Enter value that you want to search :') if field\_name =='id':

sql = 'select \* from customer where "{}" =

"{}";'.format(field\_name, value)

else:

sql = 'select \* from customer where "{}" like

"%{}%";'.format(field\_name, value)

print(sql) cursor.execute(sql) records = cursor.fetchall() clear()

print('Search Result for {} = {}'.format(field\_name,value)) print('\*'\*140)

print('{} {:20s} {:30s} {:15s} {:30s} {:20s}

{:15s}'.format('ID','Name','Address','Phone','Email','ID Used','ID No')) for record in records:

print('{} {:20s} {:30s} {:15s} {:30s} {:20s} {:15s}'.format( record[0], record[1], record[2], record[3], record[4], record[5], record[6]))

wait = input('\n\n\nPress any key to continue......')

def search\_booking():

cust\_no = input('Enter Customer No :')

sql = 'select book\_id,r.room\_no,c.name,doo,advance from booking b, customer c,rooms r where b.room\_id = r.id AND b.cust\_id = "{}" and dol is NULL;'.format(cust\_no) cursor.execute(sql) records = cursor.fetchall() clear()

print('Booking information for customer ID :{}'.format(cust\_no)) print('{} {} {} {} {}'.format('ID','RoomID', 'Customer Name','Date of Occupancy','Advance')) print('\*'\*140) for record in records:

print('{} {} {} {} {}'.format(

record[0], record[1], record[2], record[3], record[4])) wait = input('\n\n\nPress any key to continue......')

def search\_bills(): bill\_no = input('Enter Bill No :') sql = ' select bill.bill\_id,bill.amount,bill\_date,gst,st,b.book\_id,doo,dol,advance, name,address,phone,email,room\_no \

from bill, booking b, customer c , rooms r \ where bill.book\_id = b.book\_id \

and b.room\_id = r.id and b.cust\_id = c.id AND NOT dol is

NULL AND \

bill\_id = "{}";'.format(bill\_no)

cursor.execute(sql)

record = cursor.fetchone() clear()

print('Bill information for Bill No :{}'.format(bill\_no)) print('\*'\*140) print('Bill NO ', record[0]) print('Amount ', record[1]) print('Bill Date ', record[2]) print('GST Charged ', record[3]) print('Service Tax Charged ', record[4]) print('Booking ID ', record[5]) print('Room Used ID ', record[13]) print('Date of Occupancy ', record[6]) print('Date of Leaving ', record[7]) print('Advance Paid ', record[8]) print('Customer Name ', record[9]) print('Customer Address ', record[10]) print('Customer Phone ', record[11]) print('Customer Email ID ', record[12]) wait = input('\n\n\nPress any key to continue......')

def search\_menu():

while True:

clear() print(' Search Menu') print('\*'\*120) print("\n1. Room Status") print('\n2. Booking Status') print('\n3. customer Details') print('\n4. Bills') print('\n5. Back to Main Menu')

print('\n\n')

choice = int(input('Enter your choice ...: ')) if choice==1: search\_rooms() elif choice==2: search\_booking() elif choice==3:

search\_customer() elif choice==4: search\_bills() elif choice==5:

break

def report\_room\_status(): sql = 'select \* from rooms' cursor.execute(sql) records = cursor.fetchall() clear()

print(' Rooms Status - Report') print('-'\*120)

print('{:10s} {:10s} {:20s} {:20s} {:>40s} {:>30s}'.format('Room ID','Room No', 'Room Type', 'Rent','Bedding', 'Status')) for idr,no,rtype,rent,bed,status in records:

print('{:<10d} {:<10d} {:20s} {:<7.2f} {:>40s} {:>30s}'.format(idr, no, rtype, rent, bed, status))

wait = input('\n\n\n Press any key to continue....')

def report\_booking\_status():

sql = 'select b.book\_id,room\_no,doo,dol,advance, name,address,phone \

from booking b, customer c ,rooms r \

where b.room\_id = r.id and b.cust\_id = c.id and dol is NULL;' cursor.execute(sql) records = cursor.fetchall() clear()

print(' Booking Status - Report') print('-'\*120)

print('{:10s} {:10s} {:20s} {:20s} {:>30s} {:20s} {:30s} {:15s}'.format(

'Booking ID', 'Room No', 'DOO', 'DOL', 'Advance',

'Name','Address','Phone')) for idr, no, doo,dol,advance,name,addr,phone in records: print('{:10d} {:10d} {:20s} {:20s} {:10.2f} {:20s} {:30s}

{:15s}'.format(

idr, no, str(doo), str(dol), advance, name, addr, phone)) wait = input('\n\n\n Press any key to continue....')

def report\_menu(): while True:

clear()

print('Report Menu') print("\n1. Room Status") print('\n2. Booking Status') print('\n3. Back to Main Menu')

print('\n\n')

choice = int(input('Enter your choice ...: ')) if choice == 1:

report\_room\_status() elif choice == 2:

report\_booking\_status() elif choice == 3:

break

def change\_room\_status():

clear()

room\_no = input('Enter Room No :')

status = input('Enter current status(Renovation/modification ) :') sql = 'update rooms set status = "{}" where room\_no =

"{}";'.format(status, room\_no) cursor.execute(sql) print('\n\nRoom Status Updated')

wait = input('\n\n\n Press any key to continue....')

def main\_menu(): while True:

clear()

print(' H O T E L M A N A G E M E N T S Y S T E M ') print('\*'\*120)

print("\n1. Add New Room") print('\n2. Add Customer') print('\n3. Modify Room Information') print('\n4. Modify Customer Information') print('\n5. Room Booking') print('\n6. Bill Generation') print('\n7. Search Database') print('\n8. Report Menu') print('\n9. Settings') print('\n10. Close application') print('\n\n')

choice = int(input('Enter your choice ...: ')) if choice == 1: add\_room()

elif choice == 2: add\_customer() elif choice == 3: modify\_room() elif choice == 4: modify\_customer() elif choice ==5 : room\_booking() elif choice == 6: bill\_generation() elif choice ==7 : search\_menu() elif choice == 8: report\_menu() elif choice == 9: system\_settings() elif choice ==10:

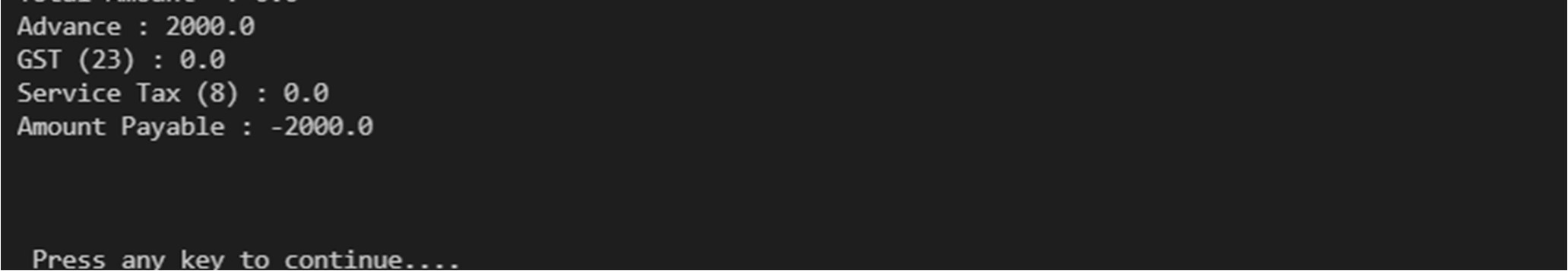
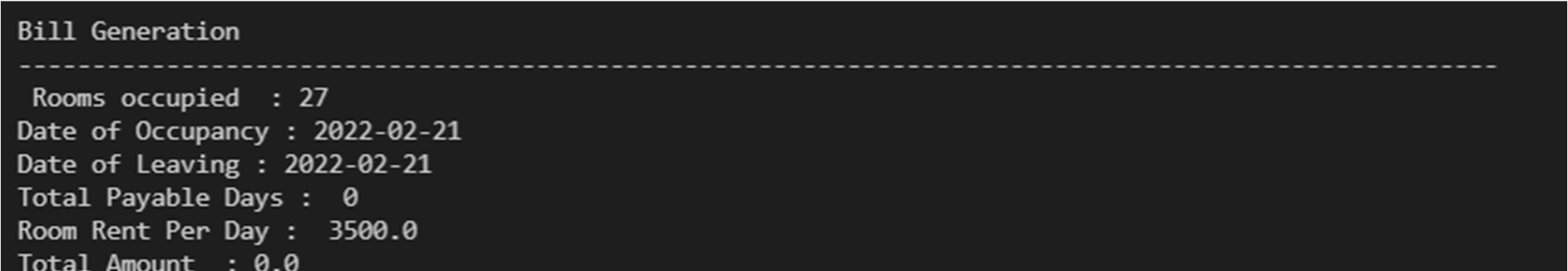
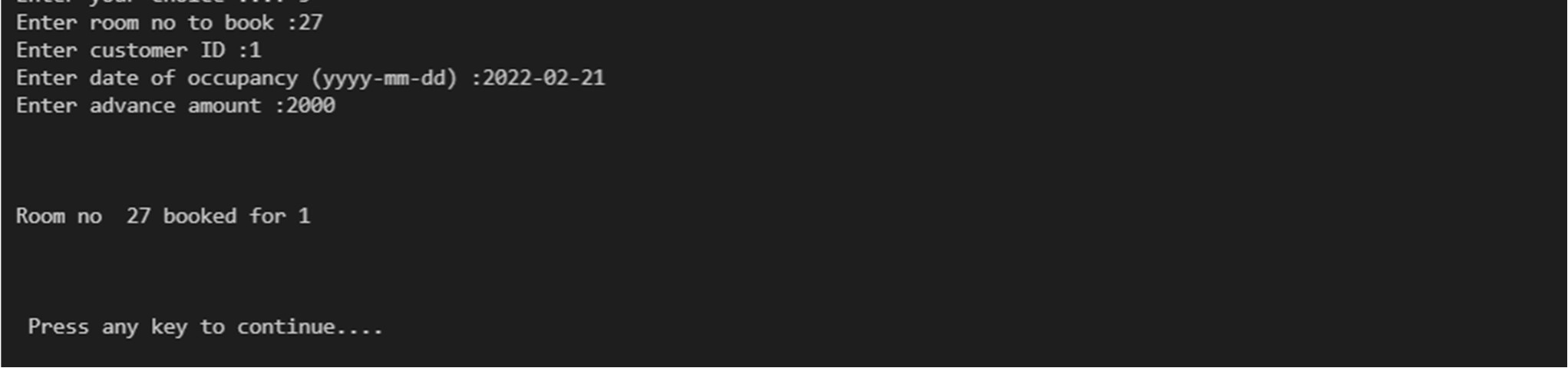
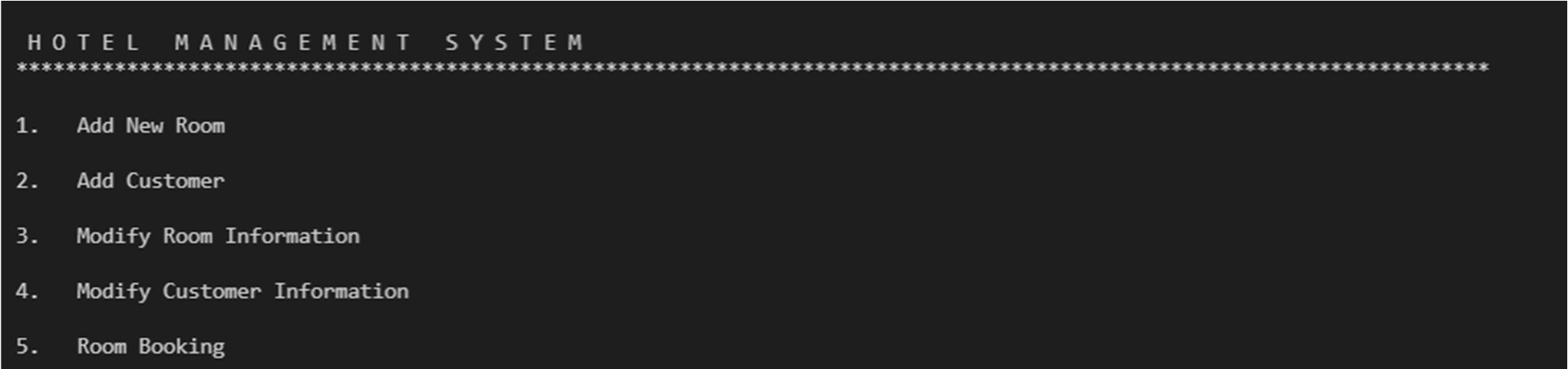
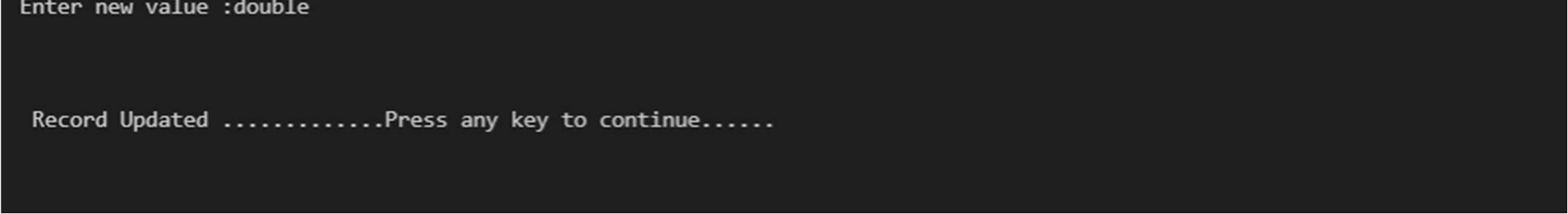
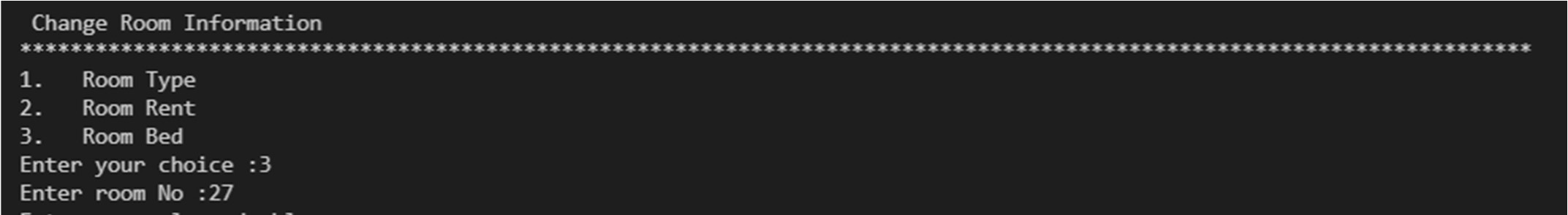
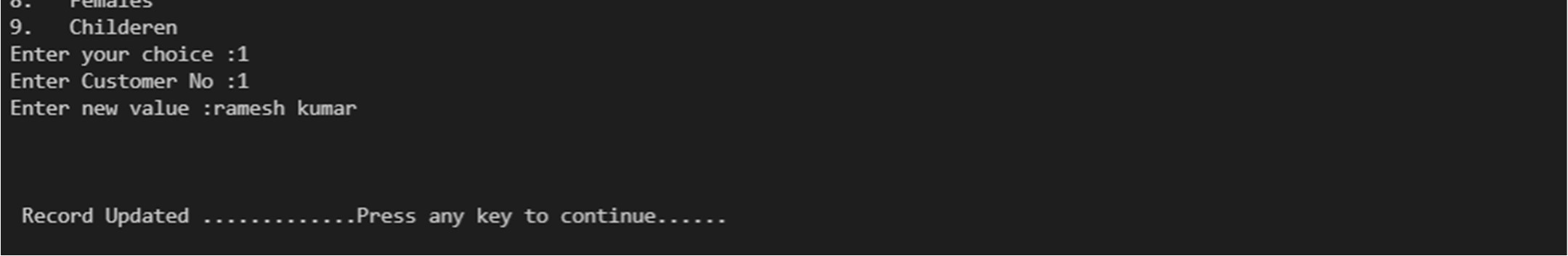
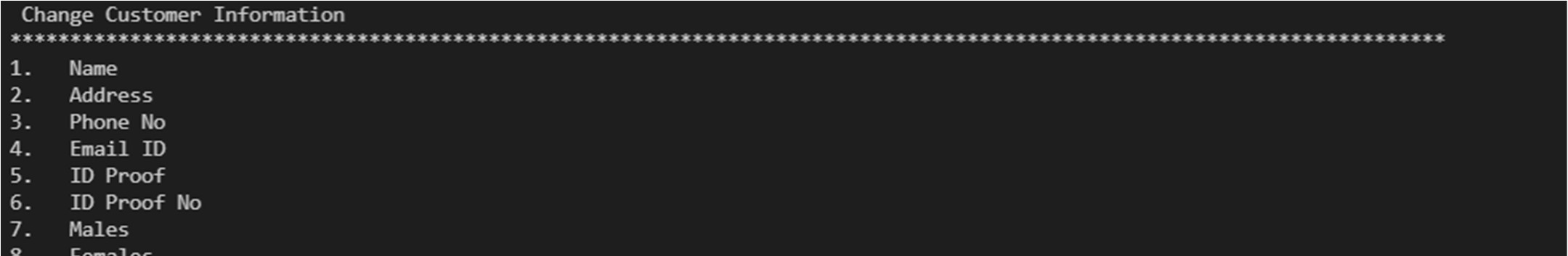
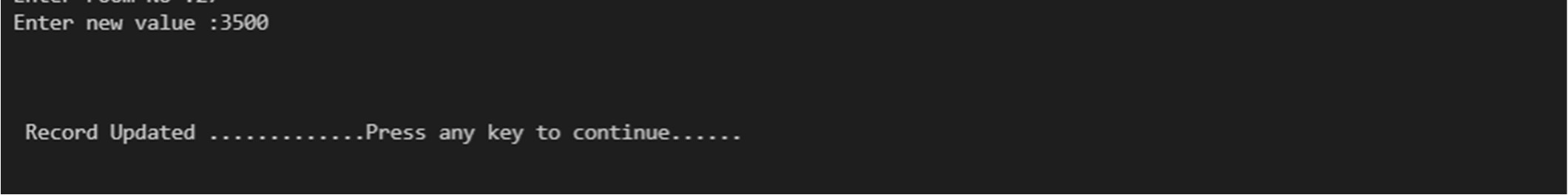
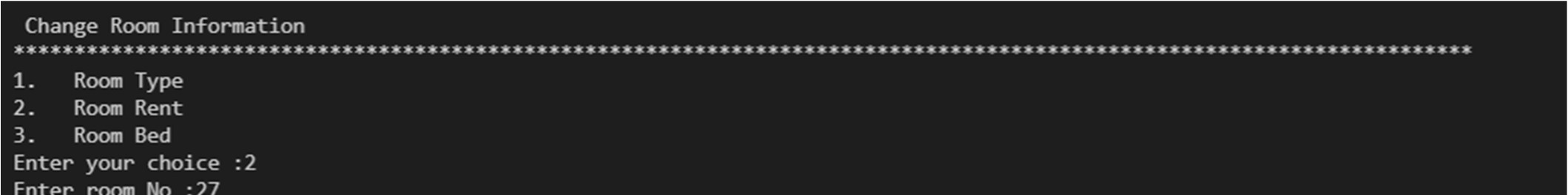
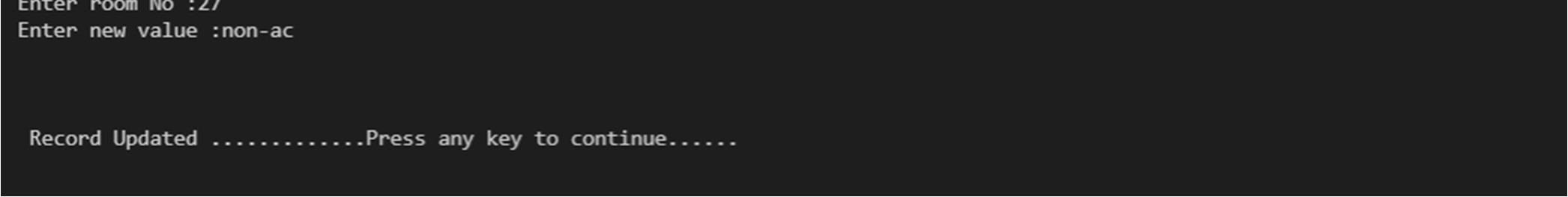
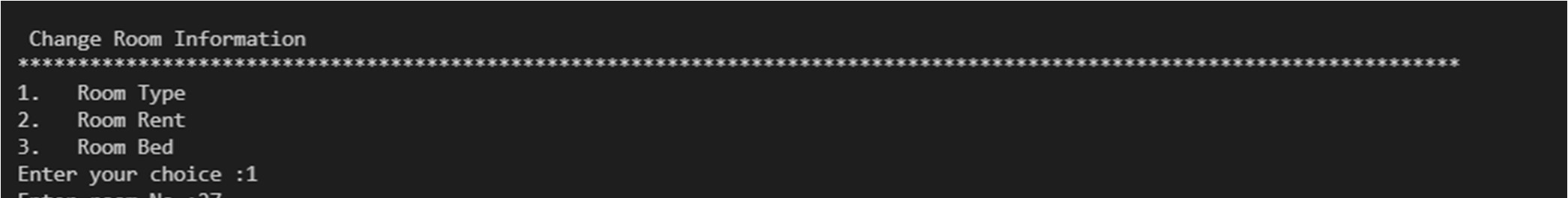
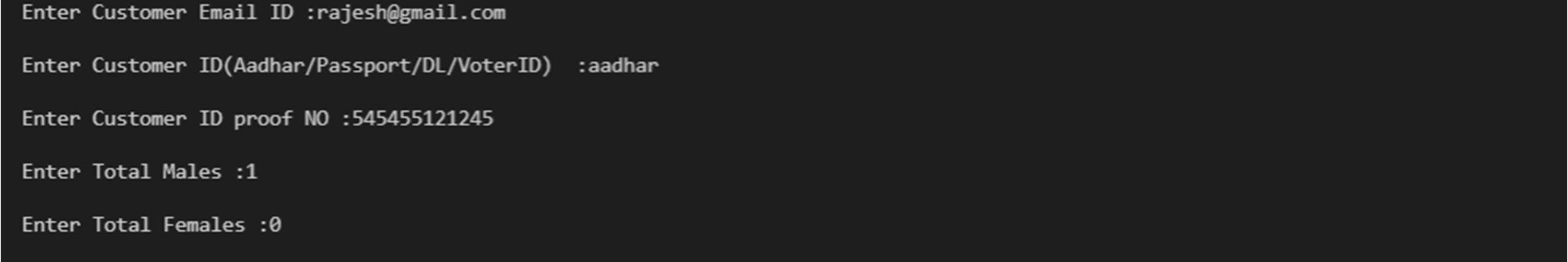
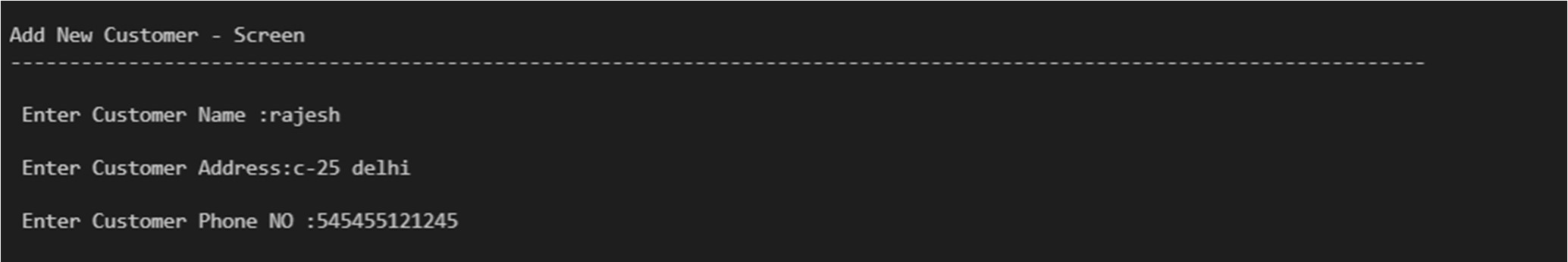
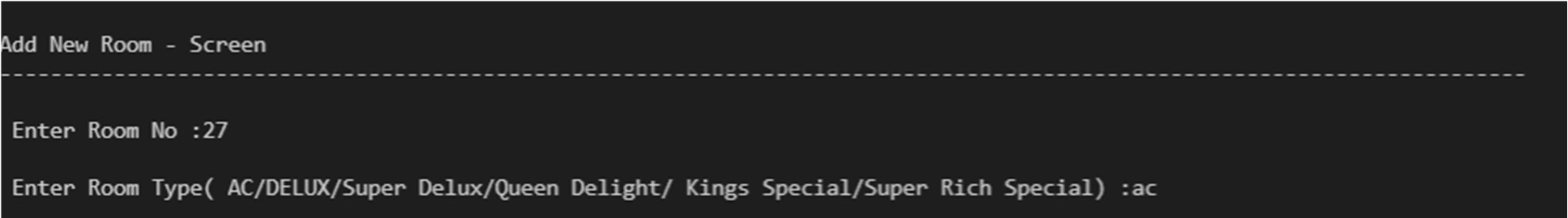
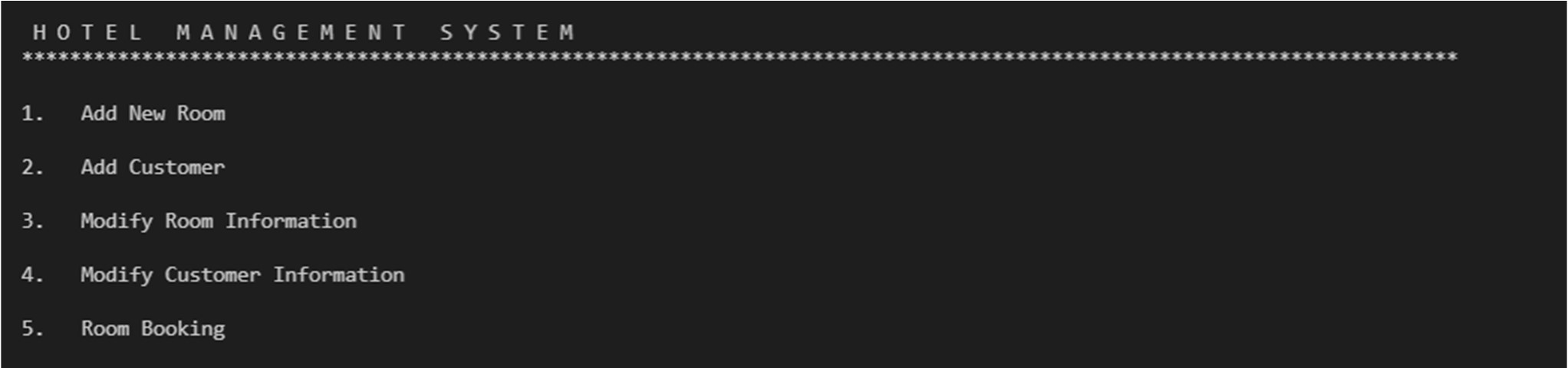
break

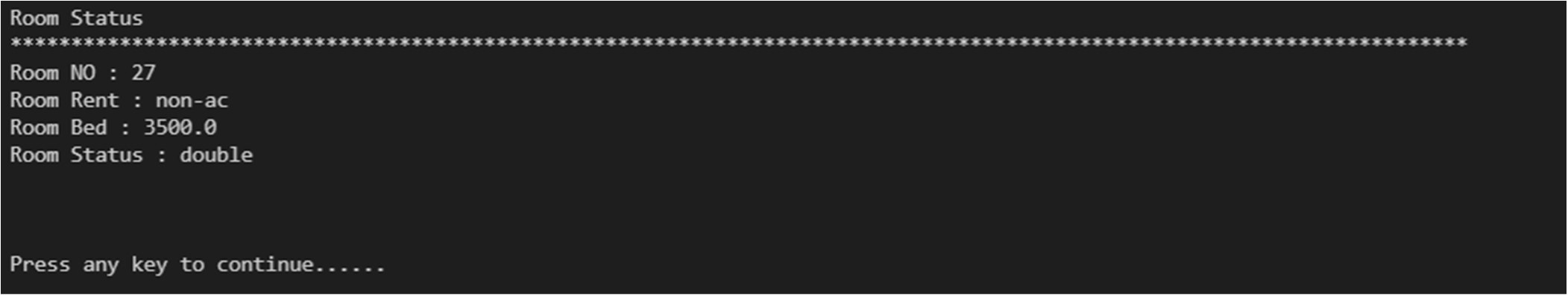
made\_by()

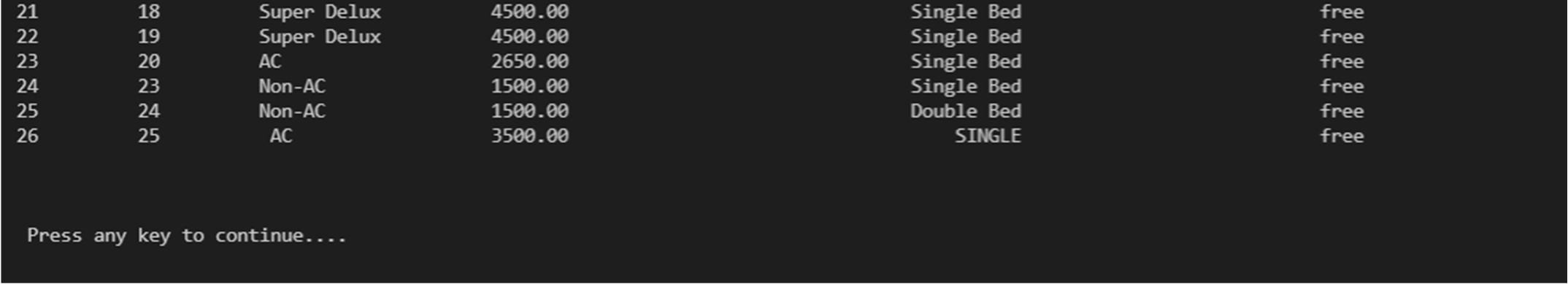
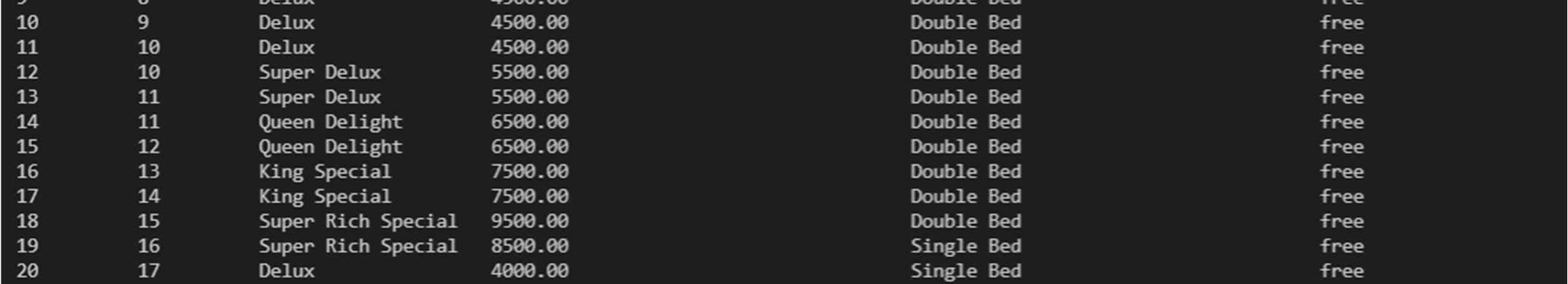
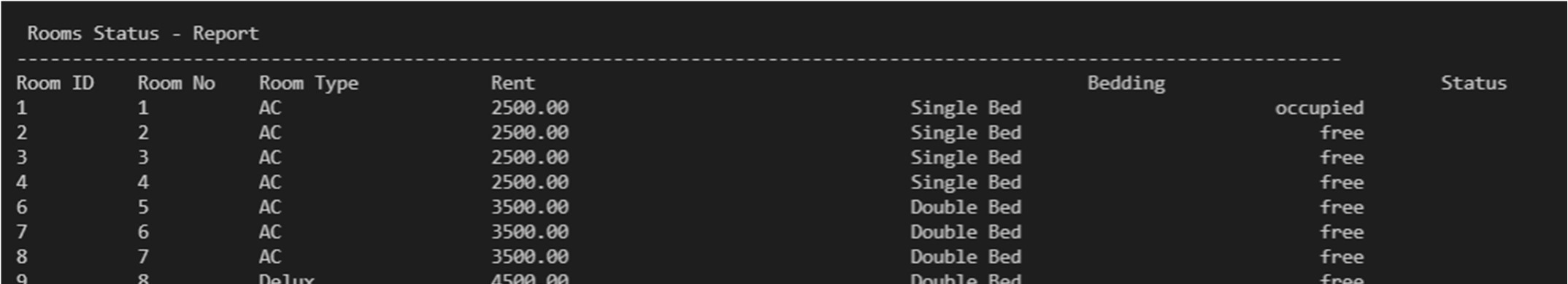
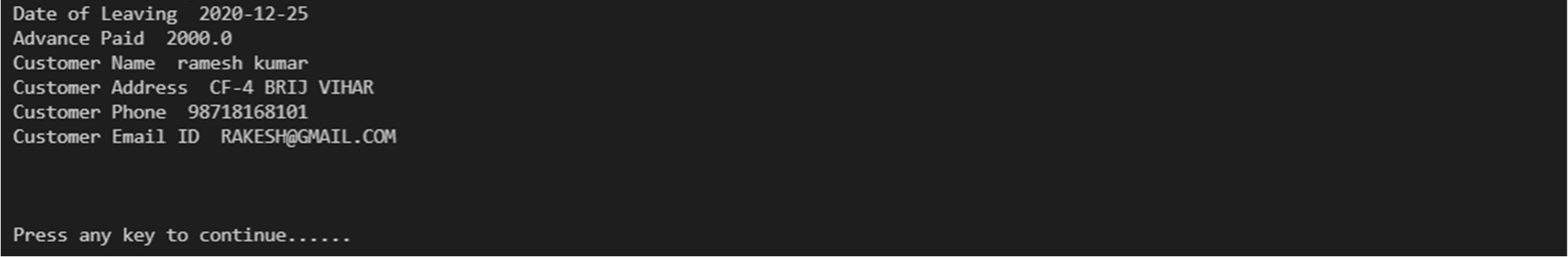
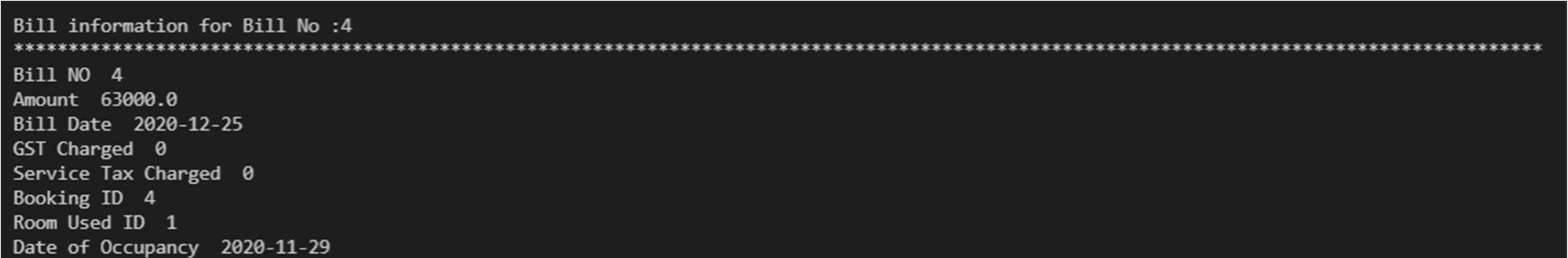
if \_\_name\_\_ == "\_\_main\_\_":

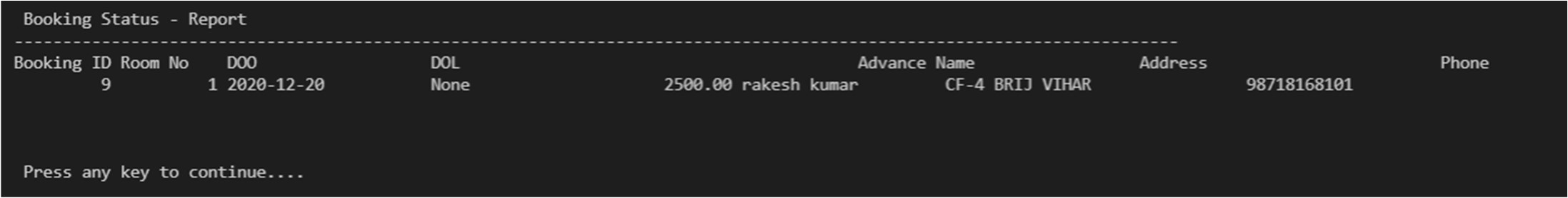
settings() main\_menu()

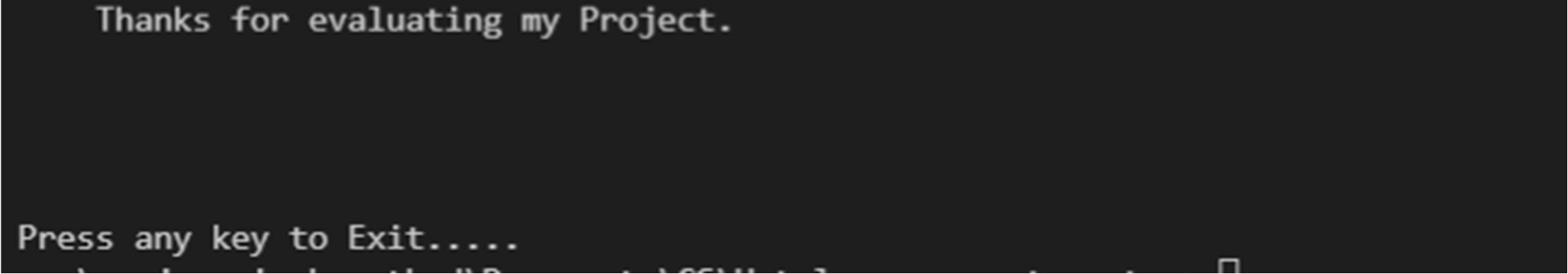
# OUTPUT SCREENSHOTS











# LIMITATIONS

1. In the proposed model fetching and updating the records is quit lengthy.
2. Special care is needed while uploading data.
3. Any mistake while uploading data may lead to again starting the process from tip.
4. It may be a bit confusing to unfamiliar users.
5. There is an uncertain in approval process.

# FUTURE SCOPE

The software to be developed deals with creating a Hotel Management system which will automate the major hotel operations such as generating COD, billing and keeping track of records of daily transaction. This project can be an idea for future projects.

# BIBLIOGRAPHY

1. Ms. Nisha Sharma , PGT CS
2. Computer science With Python - Class XI

By : Sumita Arora

1. Computer science With Python - Class XIl

By : Sumita Arora

1. Website: https://cbsetoday.com/
2. Website: https://www.w3schools.com/sql/
3. Website: https://www.geeksforgeeks.org/pythonprogramming-language/