



SHARJAH INDIAN SCHOOL , Juwaiza

Tel: 06 5665775, P.O. Box – 2324, Sharjah ; e-mail: boys@sissharjah.com Website: www.sisjuwaiza.com

COMPUTER SCIENCE PROJECT FILE HOTEL MANAGEMENT

NAME : Albin Binu Mathew

ROLL No: 05

CLASS: XII C

ACADEMIC YEAR : 2020 - 2021



SHARJAH INDIAN SCHOOL

(AFFILIATED TO CENTRAL BOARD OF SECONDARY EDUCATION, NEW DELHI)

Certificate

AISSCE Roll No : _____

Certified that the work entered in this journal on the Project

Hotel Management is the bona fide work of

Master ALBIN BINU MATHEW

of Grade XII Section C submitted for the

All India Senior School Certificate Examination conducted by

CBSE, New Delhi for the Academic Year 2020 - 2021.

Jihan Shareen
Teacher in Charge

.....
Date

.....
Internal Examiner

.....
External Examiner

CONTENTS

1.Acknowledgement

2.About Python

3.About the Project

i. Modules

ii. Functions

iii. SQL Tables

4.User Manual

5.Source Code

6.Output Screen

7.Reference Manual

8.Conclusion

9.Limitations

ACKNOWLEDGEMENT

Firstly, I would like to thank my teacher Mrs.Jihan Shareen for giving me the opportunity to do this project on the topic Hotel Management which helped me grasp a lot of new concepts and ideas .

Secondly, I would like to thank my classmate and partner Likhith for teaming up with me for this interesting project.

We both gained knowledge from this group effort.

Special thanks also go to my classmates and parents who helped me a lot in finishing this project within the limited time frame.

Albin Binu Mathew

ABOUT PYTHON

Python is an interpreted, high-level programming language created in the late 1980s and first released in 1991, by Guido Van Rossum. It is one of the most accessible programming languages available because it has a simplified syntax and is easy to learn.

Python is developed under an OSI-approved open source licence making it freely usable and distributable. Other features of python include being an integrated, portable and case-sensitive language.

It's also an interpreted language which means that Python directly executes the code line by line. In case of any error, execution of code is stopped and the error is reported.

ABOUT THE PROJECT

HOTEL MANAGEMENT

This project titled 'HOTEL MANAGEMENT' is a software to help monitor and control the overall management of a hotel.

The project is developed in Python and focuses on the basic operations in a hotel, like booking and cancelling a room and an employee access to modify and display the records.

The project also helps in monitoring the transactions in a hotel.

i. Modules

Modules are used to categorize Python code into smaller parts. A module is simply a Python file where statements, classes, objects, functions, constants and variables are defined. The Modules used in this Project are :

1. Time – Used to delay the execution of the code (time.sleep()).
2. Mysql.connector – For establishing connectivity with SQL.

ii. Functions

A function is a group of statements that exists within a program for the purpose of performing a specific task. They constitute line of code(s) that are executed sequentially from top to bottom by the Python interpreter.

Functions used in this project are:

1. **def main():** To display the Main menu.
2. **def b_room():** To book a room.
3. **def foodpurchased():** To calculate the food purchased.
4. **def cancel():** To cancel a booking.
5. **def displaypr():** To display room prices.
6. **def foodprices():** To display food prices.

7. **def crecords()**: To display customer records.
8. **def dcheck()**: To check if the user is the owner of the room.
9. **def grand()**: To calculate the grand total.
10. **def employee()**: For employee access.

iii. SQL Tables

SQL Tables enable permanent storage of the records needed in the Project.

Binary Files is used in the Project because (write advantages of Binary File)

SQL Tables used in this project are :

1. **Crecords**: Used to store all the details of customers.
2. **Employee**: Used to store employee names.
3. **F1_rooms**: Used to store the rooms available in 1st floor.
4. **F2_rooms**: Used to store the rooms available in 2nd floor.
5. **F3_rooms**: Used to store the rooms available in 3rd floor.
6. **Food**: Used to store food prices.
7. **R_prices**: Used to store room prices.

USER MANUAL

This user manual will help the user in executing this software.

❖ **Hardware Requirements:**

- X86 64-bit CPU (Intel / AMD Architecture)
- 4 GB RAM
- 5 GB free disk space

❖ **Software Requirements**

- Modern Operating System
 - Windows 7 above
 - Mac OS X 10.11 or higher, 64 bit
 - Linux : RHEL 6/7, 64 bit
 - Python 3.8 or above
- ❖ As the program is very simple the user just needs fundamental knowledge on computers.
- ❖ The installation of the program is very simple. Just insert the CD and execute the .py file using Python.

SOURCE CODE

```
1. total = 0
2. import time
3. import mysql.connector as ms
4. mydb = ms.connect(host='localhost',user='root',passwd='12345',database='hotel_management')
5. mc = mydb.cursor()
6.
7. #Booking a room
8. def b_room():
9.     ch = 'y'
10.    displaypr()
11.    print('='*40)
12.    while ch in 'Yy':
13.        print('='*40)
14.        n = input('ENTER CUSTOMER NAME: ')
15.        a = int(input('ENTER NUMBER OF ADULTS(ABOVE 12): '))
16.        k = int(input('ENTER NUMBER OF CHILDREN: '))
17.        nm = a + k
18.        if nm > 0 and nm <= 2:
19.            ci = input("ENTER CHECK-IN DATE(YYYY/MM/DD): ")
20.            co = input("ENTER CHECK-OUT DATE(YYYY/MM/DD): ")
21.            print()
22.            print("FOR",nm,",YOU HAVE BEEN ALLOTTED FIRST FLOOR ROOMS, KINDLY CHOOSE ONE OF THE BELOW...")
23.            time.sleep(1)
24.            mc.execute('select * from f1_rooms')
25.            data = mc.fetchall()
26.            print('='*40)
27.            print('=====1ST FLOOR ROOMS=====')
28.            print('='*40)
29.            if mc.rowcount>0:
30.                print('SNO', '%15s'% 'ROOM_NO', '%15s'% 'STATUS')
31.                print('='*40)
32.                for i in data:
33.                    print(i[0], '%15s'% i[1], '%20s'% i[2])
34.                    print('='*40)
35.            rno = int(input('ENTER AN AVAILABLE ROOM NUMBER: '))
36.            l = []
37.            mc.execute('select room_no from f1_rooms')
38.            data = mc.fetchall()
39.            for i in data:
40.                l.append(i[0])
41.            if rno in l:
42.                mc.execute('Select * from f1_rooms where room_no={}'.format(rno))
43.                data = mc.fetchall()
44.                status = 'BOOKED'
45.                for i in data:
46.                    if i[2] == 'AVAILABLE':
47.                        mc.execute('update f1_rooms set STATUS="{}" where ROOM_NO={}'.format(status,rno))
48.                        mydb.commit()
49.                        mc.execute('select * from f1_rooms where room_no={}'.format(rno))
50.                        data=mc.fetchall()
51.                        print('='*40)
52.                        print('=====1ST FLOOR ROOMS=====')
```

```

53.         print('-'*40)
54.         if mc.rowcount>0:
55.             print('SNO', '%14s'% 'ROOM_NO', '%17s'% 'STATUS')
56.             print('-'*40)
57.         for i in data:
58.             print(i[0], '%15s'%i[1], '%20s'%i[2])
59.             print('-'*40)
60.             print('')
61.             print('***ROOM IS SUCCESSFULLY BOOKED',n.upper(),
62. '***')
63.
64.         mc.execute("insert into crecords values({}, '{}', '{}', '{}')
        ".format(rno,n.upper(),ci,co))
65.         mydb.commit()
66.         time.sleep(1)
67.         mc.execute('select * from crecords where room_no={}'
68. .format(rno))
69.         data = mc.fetchall()
70.         if mc.rowcount>0:
71.             print('-'*50)
72.             print('ROOM NO', '%12s'% 'C_NAME', '%13s'% 'CHECK_IN',
73. '%14s'% 'CHECK_OUT')
74.             print('-'*50)
75.             for i in data:
76.                 print(i[0], '%16s'%i[1], '%13s'%i[2], '%14s'%i[3])
77.                 print('-'*50)
78.             print('GOING BACK TO MAIN MENU....')
79.             time.sleep(3)
80.             main()
81.         elif i[2] == 'BOOKED':
82.             print("-"*40)
83.             print('***ROOM IS NOT AVAILABLE, CHOOSE AN AVAILABLE
84. ONE**')
85.             ch = input('TRY AGAIN?(Y/N) ')
86.         else:
87.             print("-"*40)
88.             print("***ROOM NUMBER DOESN'T EXIST**")
89.             ch = input('TRY AGAIN?(Y/N) ')
90.
91.         elif nm <= 4 and nm > 2:
92.             ci = input("ENTER CHECK-IN DATE(YYYY/MM/DD): ")
93.             co = input("ENTER CHECK-OUT DATE(YYYY/MM/DD): ")
94.             print()
95.             print("FOR",nm,",YOU HAVE BEEN ALLOTTED SECOND FLOOR ROOMS, KINDLY CHOOSE
ONE OF THE BELOW..")
96.             time.sleep(1)
97.             mc.execute('select * from f2_rooms')
98.             data=mc.fetchall()
99.             print('-'*40)
100.             print('=====2ND FLOOR ROOMS=====')
101.             print('-'*40)
102.             if mc.rowcount>0:
103.                 print('SNO', '%15s'% 'ROOM_NO', '%15s'% 'STATUS')
104.                 print('-'*40)
105.                 for i in data:
106.                     print(i[0], '%15s'%i[1], '%20s'%i[2])
107.                     print('-'*40)
108.                 rno = int(input('ENTER AN AVAILABLE ROOM NUMBER: '))
109.                 l = []
110.                 mc.execute('select room_no from f2_rooms')
111.                 data = mc.fetchall()
112.                 for i in data:
113.                     l.append(i[0])
114.                 if rno in l:

```

```

115.                 mc.execute('Select * from f2_rooms where room_no={}'.format
116. (rno))
117.                 data = mc.fetchall()
118.                 status = 'BOOKED'
119.                 for i in data:
120.                     if i[2] == 'AVAILABLE':
121.                         mc.execute('update f2_rooms set STATUS="{}" where ROOM
122. _NO={}'.format(status,rno))
123.                         mydb.commit()
124.                         mc.execute('select * from f2_rooms where room_no={}'.
125. format(rno))
126.                         data = mc.fetchall()
127.                         print('-'*40)
128.                         if mc.rowcount>0:
129.                             print('SNO', '%14s'% 'ROOM_NO', '%17s'% 'STATUS')
130.                             print('-'*40)
131.                             for i in data:
132.                                 print(i[0], '%15s'%i[1], '%20s'%i[2])
133.                                 print('-'*40)
134.                                 print('')
135.                                 print('***ROOM IS SUCCESSFULLY BOOKED',n.upper(),
136. '***')
137.                 mc.execute("insert into crecords values({}, '{}', '{}',
138. '{}')".format(rno,n.upper(),ci,co))
139.                 mydb.commit()
140.                 time.sleep(1)
141.                 mc.execute('select * from crecords where room_no={}'.
142. format(rno))
143.                 data = mc.fetchall()
144.                 if mc.rowcount>0:
145.                     print('-'*50)
146.                     print('ROOM NO', '%12s'% 'C_NAME', '%13s'% 'CHECK_IN'
147. , '%14s'% 'CHECK_OUT')
148.                     print('-'*50)
149.                     for i in data:
150.                         print(i[0], '%16s'%i[1], '%13s'%i[2], '%14s'%i[3])
151.                         print('-'*50)
152.                         print('GOING BACK TO MAIN MENU....')
153.                         time.sleep(3)
154.                         main()
155.                     elif i[2] == 'BOOKED':
156.                         print("-"*40)
157.                         print('***ROOM IS NOT AVAILABLE, CHOOSE AN AVAILABLE
158. ONE**')
159.                         ch = input('TRY AGAIN?(Y/N) ')
160.                     else:
161.                         print("-"*40)
162.                         print('***ROOM NUMBER DOESN'T EXIST**')
163.                         ch = input('TRY AGAIN?(Y/N) ')
164.                 elif nm <= 8 and nm > 4:
165.                     ci = input("ENTER CHECK-IN DATE(YYYY/MM/DD): ")
166.                     co = input("ENTER CHECK-OUT DATE(YYYY/MM/DD): ")
167.                     print()
168.                     print("FOR",nm,",YOU HAVE BEEN ALLOTTED THIRD FLOOR ROOMS, KINDLY
169. CHOOSE ONE OF THE BELOW...")
170.                     time.sleep(1)
171.                     mc.execute('select * from f3_rooms')
172.                     data = mc.fetchall()
173.                     print('-'*40)
174.                     print('=====3RD FLOOR ROOMS=====')
175.                     print('-'*40)
176.                     if mc.rowcount>0:

```

```

170.             print('SNO', '%15s'%ROOM_NO, '%15s'%STATUS')
171.             print('-'*40)
172.         for i in data:
173.             print(i[0], '%15s'%i[1], '%20s'%i[2])
174.             print('-'*40)
175.         rno = int(input('ENTER AN AVAILABLE ROOM NUMBER: '))
176.         l = []
177.         mc.execute('select room_no from f3_rooms')
178.         data = mc.fetchall()
179.         for i in data:
180.             l.append(i[0])
181.         if rno in l:
182.             mc.execute('Select * from f3_rooms where room_no={}'.format
183. (rno))
184.             data = mc.fetchall()
185.             status = 'BOOKED'
186.             for i in data:
187.                 if i[2] == 'AVAILABLE':
188.                     mc.execute('update f3_rooms set STATUS="{}" where ROOM
189. _NO={}'.format(status,rno))
190.                     mydb.commit()
191.                     mc.execute('select * from f3_rooms where room_no={}'.
192. format(rno))
193.                     data = mc.fetchall()
194.                     print('-'*40)
195.                     if mc.rowcount>0:
196.                         print('SNO', '%14s'%ROOM_NO, '%17s'%STATUS')
197.                         print('-'*40)
198.                         for i in data:
199.                             print(i[0], '%15s'%i[1], '%20s'%i[2])
200.                             print('-'*40)
201.                             print('')
202.                             print('***ROOM IS SUCCESSFULLY BOOKED',n.upper(),
203. '***')
204.                             mc.execute("insert into crecords values({}, '{}', '{}', '
205. {}'.format(rno,n.upper(),ci,co))
206.                             mydb.commit()
207.                             time.sleep(1)
208.                             mc.execute('select * from crecords where room_no={}'.
209. format(rno))
210.                             data=mc.fetchall()
211.                             if mc.rowcount>0:
212.                                 print('-'*50)
213.                                 print('ROOM NO', '%12s'%C_NAME, '%13s'%CHECK_IN'
214. , '%14s'%CHECK_OUT')
215.                                 print('-'*50)
216.                                 for i in data:
217.                                     print(i[0], '%16s'%i[1], '%13s'%i[2], '%14s'%i[3])
218.                                     print('-'*50)
219.                                 print('GOING BACK TO MAIN MENU....')
220.                                 time.sleep(3)
221.                                 main()
222.
223.                             elif i[2] == 'BOOKED':
224.                                 print('-'*40)
225.                                 print('***ROOM IS NOT AVAILABLE, CHOOSE AN AVAILABLE
226. ONE**')
227.                                 ch = input('TRY AGAIN?(Y/N) ')
228.
229.                             else:
230.                                 print('-'*40)
231.                                 print('***ROOM NUMBER DOESN'T EXIST**')
232.                                 ch = input('TRY AGAIN?(Y/N) ')
233.
234.                             else:

```

```

226.         print('-'*40)
227.         print('**MAXIMUM CAPACITY FOR A ROOM IS 8**')
228.         ch = input('TRY AGAIN?(Y/N) ')
229.
230.     print('GOING BACK TO MAIN MENU....')
231.     time.sleep(3)
232.     main()
233.
234.     #Calculating food prices
235.     def foodpurchased():
236.         global total
237.         foodprices()
238.         print('='*40)
239.         print('6', '%17s'% 'TOTAL BILL')
240.         print('-'*40)
241.         de, be, bf, lu, di = 0, 0, 0, 0, 0
242.         c = 'y'
243.         while c in 'Yy':
244.             print('='*40)
245.             ch = int(input("ENTER THE NUMBER OF YOUR CHOICE: "))
246.             if ch == 1:
247.                 mc.execute('select * from food where sno={}'.format(ch))
248.                 data = mc.fetchall()
249.                 for i in data:
250.                     p = int(input("ENTER NUMBER OF DESSERTS: "))
251.                     de = p * i[2]
252.                     print("YOUR TOTAL DESSERT COST IS:", de)
253.
254.             elif ch == 2:
255.                 mc.execute('select * from food where sno={}'.format(ch))
256.                 data = mc.fetchall()
257.                 for i in data:
258.                     p = int(input("ENTER NUMBER OF BEVERAGES: "))
259.                     be = p * i[2]
260.                     print("YOUR TOTAL BEVERAGE COST IS:", be)
261.
262.             elif ch == 3:
263.                 mc.execute('select * from food where sno={}'.format(ch))
264.                 data = mc.fetchall()
265.                 for i in data:
266.                     p = int(input("ENTER NUMBER OF BREAKFAST MEALS: "))
267.                     bf = p * i[2]
268.                     print("YOUR TOTAL BREAKFAST COST IS:", bf)
269.
270.             elif ch == 4:
271.                 mc.execute('select * from food where sno={}'.format(ch))
272.                 data = mc.fetchall()
273.                 for i in data:
274.                     p = int(input("ENTER NUMBER OF LUNCH MEALS: "))
275.                     lu = p * i[2]
276.                     print("YOUR TOTAL LUNCH COST IS:", lu)
277.
278.             elif ch == 5:
279.                 mc.execute('select * from food where sno={}'.format(ch))
280.                 data = mc.fetchall()
281.                 for i in data:
282.                     p = int(input("ENTER NUMBER OF DINNER MEALS: "))
283.                     di = p * i[2]
284.                     print("YOUR TOTAL DINNER COST IS:", di)
285.
286.             elif ch == 6:
287.                 total = de + be + bf + lu + di
288.                 print('YOUR TOTAL FOOD COST IS', total)
289.                 print('GOING BACK TO MAIN MENU....')

```

```

290.         time.sleep(3)
291.         main()
292.
293.     else:
294.         print('-'*40)
295.         print('**INVALID CHOICE**')
296.         c = input('TRY AGAIN?(Y/N) ')
297.
298.     print('GOING BACK TO MAIN MENU....')
299.     time.sleep(3)
300.     main()
301.
302. #cancelling a room
303. def cancel():
304.     ch = 'y'
305.     while ch in 'Yy':
306.         print('-'*40)
307.         fno = int(input('ENTER YOUR FLOOR NUMBER(1/2/3): '))
308.         if fno == 1:
309.             rno = int(input('ENTER ROOM NUMBER TO BE CANCELLED: '))
310.             l1 = []
311.             mc.execute('select room_no from f1_rooms')
312.             data = mc.fetchall()
313.             for i in data:
314.                 l1.append(i[0])
315.             if rno in l1:
316.                 mc.execute('select * from f1_rooms where room_no={}'.format(rno)
317.                 )
318.                 data = mc.fetchall()
319.                 status = 'AVAILABLE'
320.                 for i in data:
321.                     if i[2] == 'BOOKED':
322.                         mc.execute("update f1_rooms set status='{}' where room
323. _no={}".format(status,rno))
324.                         mydb.commit()
325.                         mc.execute('select * from f1_rooms where room_no={}'.
326. format(rno))
327.                         data = mc.fetchall()
328.                         print('-'*40)
329.                         print('=====1ST FLOOR ROOMS=====')
330.                         print('-'*40)
331.                         if mc.rowcount>0:
332.                             print('SNO', '%14s'%'ROOM_NO', '%17s'
333. %'STATUS')
334.                             print('-'*40)
335.                             for i in data:
336.                                 print(i[0], '%15s'%'i[1], '%20s'%'i[2])
337.                                 print('-'*40)
338.                             mc.execute('delete from crecords where room_no={}'.
339. format(rno))
340.                             mydb.commit()
341.                             print('**ROOM HAS BEEN CANCELLED**')
342.                             print('GOING BACK TO MAIN MENU....')
343.                             time.sleep(3)
344.                             main()
345.
346.                     elif i[2] == 'AVAILABLE':
347.                         print('-'*40)
348.                         print('**ROOM HAS NEVER BEEN BOOKED**')
349.                         ch = input('TRY AGAIN?(Y/N)? ')
350.
351.     else:
352.         print('-'*40)
353.         print('**ROOM NO DOESN'T EXIST**')
354.         ch=input('TRY AGAIN?(Y/N) ')

```

```

349.
350.         elif fno == 2:
351.             rno = int(input('ENTER ROOM NUMBER TO BE CANCELLED: '))
352.             l1 = []
353.             mc.execute('select room_no from f2_rooms')
354.             data = mc.fetchall()
355.             for i in data:
356.                 l1.append(i[0])
357.             if rno in l1:
358.                 mc.execute('select * from f2_rooms where room_no={}'.format
359. (rno))
360.                 data = mc.fetchall()
361.                 status = 'AVAILABLE'
362.                 for i in data:
363.                     if i[2] == 'BOOKED':
364.                         mc.execute("update f2_rooms set status='{}' where room
365. _no={}".format(status,rno))
366.                         mydb.commit()
367.                         mc.execute('select * from f2_rooms where room_no={}'.
368. format(rno))
369.                         data = mc.fetchall()
370.                         print('-'*40)
371.                         print('=====2ND FLOOR ROOMS=====')
372.                         print('-'*40)
373.                         if mc.rowcount>0:
374.                             print('SNO', '%14s'%'ROOM_NO', '%17s'%'STATUS')
375.                             print('-'*40)
376.                             for i in data:
377.                                 print(i[0], '%15s'%i[1], '%20s'%i[2])
378.                                 print('-'*40)
379.                             mc.execute('delete from crecords where room_no={}'.
380. format(rno))
381.                             mydb.commit()
382.                             print('**ROOM HAS BEEN CANCELLED**')
383.                             print('GOING BACK TO MAIN MENU....')
384.                             time.sleep(3)
385.                             main()
386.
387.                     elif i[2] == 'AVAILABLE':
388.                         print('-'*40)
389.                         print('**ROOM HAS NEVER BEEN BOOKED**')
390.                         ch=input('TRY AGAIN?(Y/N) ')
391.
392.                 else:
393.                     print('-'*40)
394.                     print("***ROOM NO DOESN'T EXIST**")
395.                     ch = input('TRY AGAIN?(Y/N) ')
396.
397.             elif fno == 3:
398.                 rno = int(input('ENTER ROOM NUMBER TO BE CANCELLED: '))
399.                 l1 = []
400.                 mc.execute('select room_no from f3_rooms')
401.                 data = mc.fetchall()
402.                 for i in data:
403.                     l1.append(i[0])
404.                 if rno in l1:
405.                     mc.execute('select * from f3_rooms where room_no={}'.format
406. (rno))
407.                     data = mc.fetchall()
408.                     status = 'AVAILABLE'
409.                     for i in data:
410.                         if i[2] == 'BOOKED':
411.                             mc.execute("update f3_rooms set status='{}' where room
412. _no={}".format(status,rno))
413.                             mydb.commit()

```

```

407.         mc.execute('select * from f3_rooms where room_no={}'.
format(rno))
408.         data = mc.fetchall()
409.         print('-'*40)
410.         print('=====3RD FLOOR ROOMS=====')
411.         print('-'*40)
412.         if mc.rowcount>0:
413.             print('SNO', '%14s'% 'ROOM_NO', '%17s'% 'STATUS')
414.             print('-'*40)
415.             for i in data:
416.                 print(i[0], '%15s'%i[1], '%20s'%i[2])
417.                 print('-'*40)
418.         mc.execute('delete from crecords where room_no={}'.
format(rno))
419.         mydb.commit()
420.         print('**ROOM HAS BEEN CANCELLED**')
421.         print('GOING BACK TO MAIN MENU...')
422.         time.sleep(3)
423.         main()
424.
425.         elif i[2] == 'AVAILABLE':
426.             print('-'*40)
427.             print('**ROOM HAS NEVER BEEN BOOKED**')
428.             ch = input('TRY AGAIN?(Y/N) ')
429.
430.         else:
431.             print('-'*40)
432.             print("***ROOM NO DOESN'T EXIST**")
433.             ch = input('TRY AGAIN?(Y/N) ')
434.
435.         else:
436.             print('-'*40)
437.             print('**INVALID FLOOR**')
438.             ch = input('TRY AGAIN?(Y/N) ')
439.
440.         print('GOING BACK TO MAIN MENU...')
441.         time.sleep(3)
442.         main()
443.
444. #displaying customer records
445. def crecords():
446.     mc.execute('select * from crecords')
447.     data = mc.fetchall()
448.     print('-'*50)
449.     print('=====DISPLAYING CUSTOMER RECORDS=====')
450.     print('-'*50)
451.     time.sleep(1)
452.     if mc.rowcount>0:
453.         print('ROOM NO', '%12s'% 'C_NAME', '%13s'% 'CHECK_IN', '%14s'% 'CHECK_OUT')
454.         print('-'*50)
455.         for i in data:
456.             print(i[0], '%16s'%i[1], '%13s'%i[2], '%14s'%i[3])
457.             print('-'*50)
458.
459. #displaying room prices
460. def displaypr():
461.     mc.execute('select * from r_prices')
462.     data=mc.fetchall()
463.     print('-'*40)
464.     print('=====PRICES PER NIGHT (INR)=====')
465.     print('-'*40)
466.     if mc.rowcount>0:
467.         print('SNO', '%15s'% 'FLOORS', '%15s'% 'PRICE')
468.         print('-'*40)
469.         for i in data:

```



```

468.         print(i[0], '%19s'%i[1], '%12s'%i[2])
469.         print('-'*40)
470.
471.     #displaying food prices
472.     def foodprices():
473.         mc.execute('select * from food')
474.         data = mc.fetchall()
475.         print('-'*40)
476.         print('=====HOTEL MENU=====')
477.         print('-'*40)
478.         if mc.rowcount>0:
479.             print('SNO', '%13s'% 'MENU', '%16s'% 'PRICE')
480.             print('-'*40)
481.             for i in data:
482.                 print(i[0], '%17s'%i[1], '%13s'%i[2])
483.                 print('-'*40)
484.
485.     def dcheck():
486.         ch = 'y'
487.         while ch in 'Yy':
488.             print('*'*40)
489.             rno = int(input('ENTER YOUR ROOM NUMBER: '))
490.             n = input('ENTER YOUR NAME AS GIVEN IN RECORD: ')
491.             mc.execute('select room_no from crecords')
492.             data = mc.fetchall()
493.             list1 = []
494.             for i in data:
495.                 list1.append(i[0])
496.             if rno in list1:
497.                 mc.execute('select * from crecords where room_no={}'.format(rno))
498.                 data = mc.fetchall()
499.                 for i in data:
500.                     if i[1] == n.upper():
501.                         print('ROOM FOUND...')
502.                         print('-'*40)
503.                         time.sleep(2)
504.                         grand()
505.                     else:
506.                         print('-'*40)
507.                         print('***CUSTOMER RECORD NOT FOUND**')
508.                         ch=input('TRY AGAIN?(Y/N) ')
509.             else:
510.                 print('-'*40)
511.                 print('***ROOM NUMBER IS AVAILABLE/DOESN'T EXIST**')
512.                 ch=input('TRY AGAIN?(Y/N) ')
513.
514.         print('GOING BACK TO MAIN MENU....')
515.         time.sleep(3)
516.         main()
517.
518.     #Calculating grand total
519.     def grand():
520.         ch = 'y'
521.         while ch in 'Yy':
522.             fno = int(input('ENTER YOUR FLOOR NUMBER: '))
523.             if fno in (1,2,3):
524.                 mc.execute('select * from r_prices where sno={}'.format(fno))
525.                 data = mc.fetchall()
526.                 nn = int(input('NUMBER OF NIGHTS YOU HAVE STAYED: '))
527.                 for i in data:
528.                     print('-'*40)
529.                     print('----DISPLAYING YOUR PRICE----')
530.                     time.sleep(2)
531.                     print('-'*40)

```

```

532.         print('YOUR ROOM PRICE IS',i[2] * nn)
533.         print('-'*40)
534.         print('YOUR TOTAL FOOD COST IS:',total)
535.         print('-'*40)
536.         time.sleep(2)
537.         gt= (i[2] * nn) + total
538.         print('YOUR GRAND TOTAL IS: ',gt)
539.         print('-'*40)
540.         print('**THANK YOU FOR YOUR STAY, VISIT AGAIN**')
541.         print('-'*40)
542.         time.sleep(2)
543.         exit()
544.     else:
545.         print("-"*40)
546.         print('**INVALID NUMBER GIVEN**')
547.         ch=input('TRY AGAIN?(Y/N) ')
548.
549.         print('GOING BACK TO MAIN MENU....')
550.         time.sleep(3)
551.         main()
552.
553. #Employee access
554. def employee():
555.     d = 12345
556.     ch = 'y'
557.     while ch == 'y':
558.         print('='*40)
559.         ena = input('ENTER EMPLOYEE NAME: ')
560.         l = []
561.         mc.execute('select emp_name from employee')
562.         data = mc.fetchall()
563.         for i in data:
564.             l.append(i[0])
565.         if ena.upper() in l:
566.             pw = int(input('ENTER PASSWORD: '))
567.             if pw == d:
568.                 print('*'*40)
569.                 print('WELCOME TO HOTEL EMPLOYEE ACCESS',ena.upper())
570.                 break
571.             else:
572.                 print('*'*40)
573.                 print('**INVALID PASSWORD**')
574.                 i=input('TRY AGAIN?(Y/N) ')
575.                 if i in 'Nn':
576.                     print('GOING BACK TO MAIN MENU....')
577.                     time.sleep(2)
578.                     main()
579.             else:
580.                 print("*"*40)
581.                 print('**EMPLOYEE NOT FOUND**')
582.                 e = input('TRY AGAIN?(Y/N) ')
583.                 if e in 'Nn':
584.                     print('GOING BACK TO MAIN MENU....')
585.                     time.sleep(2)
586.                     main()
587.         while True:
588.             time.sleep(1)
589.             print('='*50)
590.             print("""TO MODIFY ROOM PRICES, ENTER 1
591. TO MODIFY FOOD PRICES, ENTER 2
592. TO DISPLAY CUSTOMER RECORDS,ENTER 3
593. TO EXIT EMPLOYEE ACCESS,ENTER 4""")
594.             m=int(input("ENTER YOUR CHOICE: "))
595.

```

```

596.         if m == 1:
597.             print('*'*40)
598.             print("DISPLAYING ROOM PRICES.....")
599.             time.sleep(2)
600.             print()
601.             displaypr()
602.             time.sleep(1)
603.             ch = 'y'
604.             while ch in 'Yy':
605.                 print('*'*40)
606.                 print("""ENTER 1 TO MODIFY FIRST FLOOR
607. ENTER 2 TO MODIFY SECOND FLOOR
608. ENTER 3 TO MODIFY THIRD FLOOR""")
609.                 print('*'*40)
610.                 fno=int(input('ENTER FLOOR NUMBER: '))
611.                 if fno in (1,2,3):
612.                     mc.execute('select price from r_prices where sno={}'.format
(fno))
613.                     data = mc.fetchall()
614.                     price = int(input('ENTER NEW PRICE: '))
615.                     for i in data:
616.                         if price != i[0]:
617.                             mc.execute('update r_prices set PRICE={} where
sno={}'.format(price,fno))
618.                             mydb.commit()
619.                             mc.execute('select * from r_prices where sno={}'.
format(fno))
620.                             data = mc.fetchall()
621.                             print('*'*40)
622.                             print("=====PRICES CHANGED=====")
623.                             print('*'*40)
624.                             if mc.rowcount>0:
625.                                 print('SNO', '%14s'% 'FLOORS', '%17s'% 'PRICE')
626.                                 print('*'*40)
627.                                 for i in data:
628.                                     print(i[0], '%15s'%i[1], '%20s'%i[2])
629.                                     print('*'*40)
630.                                 ch = input('CHANGE ANOTHER PRICE?(Y/N) ')
631.                                 if ch in 'Yy':
632.                                     break
633.                                 elif ch in 'Nn':
634.                                     print('*'*40)
635.                                     print('GOING BACK TO EMPLOYEE ACCESS....')
636.                                     time.sleep(2)
637.
638.                                 elif price == i[0]:
639.                                     print('*'*40)
640.                                     print('**THE PRICE ENTERED IS SAME**')
641.                                     print('**TRY AGAIN**')
642.                                     time.sleep(2)
643.                                     break
644.                             else:
645.                                 print('*'*40)
646.                                 print('**INVALID FLOOR ENTERED**')
647.                                 ch=input('TRY AGAIN?(Y/N) ')
648.                                 if ch in 'Nn':
649.                                     print('GOING BACK TO EMPLOYEE ACCESS....')
650.                                     time.sleep(2)
651.
652.             elif m==2:
653.                 print('*'*40)
654.                 print('DISPLAYING FOOD PRICES.....')

```

```

655.         time.sleep(2)
656.         print()
657.         foodprices()
658.         time.sleep(1)
659.         ch = 'y'
660.         while ch in 'Yy':
661.             print('-'*40)
662.             print("""ENTER 1 TO MODIFY DESSERT PRICES
663.     ENTER 2 TO MODIFY BEVERAGE PRICES
664.     ENTER 3 TO MODIFY BREAKFAST PRICES
665.     ENTER 4 TO MODIFY LUNCH PRICES
666.     ENTER 5 TO MODIFY DINNER PRICES""")
667.             print('-'*40)
668.             fono=int(input('ENTER FOOD CHOICE: '))
669.             if fono in (1,2,3,4,5):
670.                 mc.execute('select price from food where sno={}'.format
671. (fono))
672.                 data = mc.fetchall()
673.                 price = int(input('ENTER NEW PRICE: '))
674.                 for i in data:
675.                     if price != i[0]:
676.                         mc.execute('update food set price={} where sno={}'
677. .format(price,fono))
678.                         mydb.commit()
679.                         mc.execute('select * from food where sno={}'
680. .format(fono))
681.                         data = mc.fetchall()
682.                         print('-'*40)
683.                         print('=====PRICES CHANGED=====')
684.                         print('-'*40)
685.                         if mc.rowcount>0:
686.                             print('SNO', '%18s'% 'MENU', '%17s'% 'PRICE')
687.                             print('-'*40)
688.                             for i in data:
689.                                 print(i[0], '%20s'%i[1], '%17s'%i[2])
690.                                 print('-'*40)
691.                                 ch = input('CHANGE ANOTHER PRICE?(Y/N) ')
692.                                 if ch in 'Yy':
693.                                     break
694.                                 elif ch in 'Nn':
695.                                     print('GOING BACK TO EMPLOYEE ACCESS....')
696.                                     time.sleep(2)
697.                                     elif price == i[0]:
698.                                         print('-'*40)
699.                                         print('**THE PRICE ENTERED IS SAME**')
700.                                         print('**TRY AGAIN**')
701.                                         time.sleep(2)
702.                                         break
703.                                     else:
704.                                         print("-"*40)
705.                                         print('**INVALID CHOICE**')
706.                                         ch=input('TRY AGAIN?(Y/N) ')
707.                                         if ch in 'Nn':
708.                                             print('GOING BACK TO EMPLOYEE ACCESS....')
709.                                             time.sleep(2)
710.                                             elif m==3:
711.                                                 crecords()
712.                                             elif m==4:
713.                                                 print('-'*40)
714.                                                 print('EXITING EMPLOYEE ACCESS...')

```

```

715.         time.sleep(2)
716.         main()
717.
718.     else:
719.         print('-'*40)
720.         print('**YOU HAVE ENTERED AN INVALID OPTION**')
721.         ch=input('TRY AGAIN?(Y/N) ')
722.         if ch in 'Nn':
723.             print('GOING BACK TO MAIN MENU...')
724.             time.sleep(2)
725.             main()
726. #Main program
727. def main():
728.
729.     try:
730.         print('-'*40)
731.         print('=====MENU=====')
732.         print('-'*40)
733.         print("TO BOOK A ROOM, ENTER 1")
734.         print('-'*40)
735.         print("TO PURCHASE FOOD ITEMS,ENTER 2")
736.         print('-'*40)
737.         print("TO CANCEL ROOM,ENTER 3")
738.         print('-'*40)
739.         print("TO DISPLAY GRAND TOTAL,ENTER 4")
740.         print('-'*40)
741.         print("FOR EMPLOYEE ACCESS,ENTER 5")
742.         print('-'*40)
743.         print("TO EXIT,ENTER 6")
744.         print('-'*40)
745.         c = int(input('ENTER YOUR CHOICE: '))
746.
747.         if c == 1:
748.             b_room()
749.
750.         elif c == 2:
751.             #total = 0
752.             foodpurchased()
753.
754.         elif c == 3:
755.             cancel()
756.
757.         elif c == 4:
758.             dcheck()
759.
760.         elif c == 5:
761.             employee()
762.
763.         elif c == 6:
764.             print('*'*20)
765.             print('EXITING...')
766.             time.sleep(2)
767.             exit()
768.
769.     else:
770.         print('**YOU HAVE ENTERED AN INVALID CHOICE**')
771.         time.sleep(1)
772.         main()
773.
774. except:
775.     print('*'*64)
776.     print('**YOU HAVE ENTERED AN INVALID CHARACTER, PLEASE ENTER A NUMBER**')
777.
778.     print('*'*64)

```

```

778.         time.sleep(1)
779.         main()
780.     print('*'*40)
781.     print("-----WELCOME TO HOTEL VIENNA-----")
782.     print('*'*40)
783.     while True:
784.         main()

```

OUTPUT SCREEN

1) BOOKING A ROOM

```

*****
-----WELCOME TO HOTEL VIENNA-----
*****

=====MENU=====

TO BOOK A ROOM, ENTER 1
-----
TO PURCHASE FOOD ITEMS,ENTER 2
-----
TO CANCEL ROOM,ENTER 3
-----
TO DISPLAY GRAND TOTAL,ENTER 4
-----
FOR EMPLOYEE ACCESS,ENTER 5
-----
TO EXIT,ENTER 6
-----
ENTER YOUR CHOICE: 1
-----
=====PRICES PER NIGHT (INR)=====
-----
SNO          FLOORS          PRICE
-----
1            FIRST FLOOR      800
-----
2            SECOND FLOOR     1500
-----
3            THIRD FLOOR      2200
-----
=====
=====
ENTER CUSTOMER NAME: albin binu
ENTER NUMBER OF ADULTS (ABOVE 12): 2
ENTER NUMBER OF CHILDREN: 2
ENTER CHECK-IN DATE (YYYY/MM/DD): 2021/02/15
ENTER CHECK-OUT DATE (YYYY/MM/DD): 2021/02/18

```

FOR 4 ,YOU HAVE BEEN ALLOTTED SECOND FLOOR ROOMS, KINDLY CHOOSE ONE OF THE BELOW...

=====2ND FLOOR ROOMS=====

SNO	ROOM_NO	STATUS
1	201	AVAILABLE
2	202	BOOKED
3	203	BOOKED
4	204	AVAILABLE
5	205	AVAILABLE
6	206	AVAILABLE
7	207	BOOKED
8	208	AVAILABLE

ENTER AN AVAILABLE ROOM NUMBER: 205

SNO	ROOM_NO	STATUS
5	205	BOOKED

***ROOM IS SUCCESSFULLY BOOKED ALBIN BINU ***

ROOM_NO	C_NAME	CHECK_IN	CHECK_OUT
205	ALBIN BINU	2021-02-15	2021-02-18

GOING BACK TO MAIN MENU....

2) PURCHASING FOOD

=====MENU=====

TO BOOK A ROOM, ENTER 1

TO PURCHASE FOOD ITEMS, ENTER 2

TO CANCEL ROOM, ENTER 3

TO DISPLAY GRAND TOTAL, ENTER 4

FOR EMPLOYEE ACCESS, ENTER 5

TO EXIT, ENTER 6

ENTER YOUR CHOICE: 2

=====HOTEL MENU=====

SNO	MENU	PRICE
1	DESSERT	120
2	BEVERAGES	50
3	BREAKFAST	150
4	LUNCH	150
5	DINNER	220
6	TOTAL BILL	

ENTER THE NUMBER OF YOUR CHOICE: 1

ENTER NUMBER OF DESSERTS: 2

YOUR TOTAL DESSERT COST IS: 240

ENTER THE NUMBER OF YOUR CHOICE: 4

ENTER NUMBER OF LUNCH MEALS: 3

YOUR TOTAL LUNCH COST IS: 450

ENTER THE NUMBER OF YOUR CHOICE: 5

ENTER NUMBER OF DINNER MEALS: 2

YOUR TOTAL DINNER COST IS: 440

ENTER THE NUMBER OF YOUR CHOICE: 2

ENTER NUMBER OF BEVERAGES: 2

YOUR TOTAL BEVERAGE COST IS: 100

ENTER THE NUMBER OF YOUR CHOICE: 6

YOUR TOTAL FOOD COST IS 1230

GOING BACK TO MAIN MENU....

3) CANCELLING ROOM

=====MENU=====

TO BOOK A ROOM, ENTER 1

TO PURCHASE FOOD ITEMS, ENTER 2

TO CANCEL ROOM, ENTER 3

TO DISPLAY GRAND TOTAL, ENTER 4

FOR EMPLOYEE ACCESS, ENTER 5

TO EXIT, ENTER 6

ENTER YOUR CHOICE: 3

ENTER YOUR FLOOR NUMBER (1/2/3): 2

ENTER ROOM NUMBER TO BE CANCELLED: 205

=====2ND FLOOR ROOMS=====

SNO	ROOM_NO	STATUS
5	205	AVAILABLE

ROOM HAS BEEN CANCELLED
GOING BACK TO MAIN MENU....

4) DISPLAYING GRAND TOTAL

=====MENU=====

TO BOOK A ROOM, ENTER 1

TO PURCHASE FOOD ITEMS, ENTER 2

TO CANCEL ROOM, ENTER 3

TO DISPLAY GRAND TOTAL, ENTER 4

FOR EMPLOYEE ACCESS, ENTER 5

TO EXIT, ENTER 6

ENTER YOUR CHOICE: 4

ENTER YOUR ROOM NUMBER: 205

ENTER YOUR NAME AS GIVEN IN RECORD: ALBIN BINU
ROOM FOUND...

ENTER YOUR FLOOR NUMBER: 2

NUMBER OF NIGHTS YOU HAVE STAYED: 3

----DISPLAYING YOUR PRICE----

YOUR ROOM PRICE IS 4500

YOUR TOTAL FOOD COST IS: 2250

YOUR GRAND TOTAL IS: 6750

THANK YOU FOR YOUR STAY, VISIT AGAIN

5) EMPLOYEE ACCESS

```
*****
-----WELCOME TO HOTEL VIENNA-----
*****

=====MENU=====

TO BOOK A ROOM, ENTER 1
-----
TO PURCHASE FOOD ITEMS, ENTER 2
-----
TO CANCEL ROOM, ENTER 3
-----
TO DISPLAY GRAND TOTAL, ENTER 4
-----
FOR EMPLOYEE ACCESS, ENTER 5
-----
TO EXIT, ENTER 6
-----
ENTER YOUR CHOICE: 5
=====
ENTER EMPLOYEE NAME: eric
ENTER PASSWORD: 12345
*****
WELCOME TO HOTEL EMPLOYEE ACCESS ERIC
=====
```

5.1) MODIFYING ROOM PRICES

```
TO MODIFY ROOM PRICES, ENTER 1
TO MODIFY FOOD PRICES, ENTER 2
TO DISPLAY CUSTOMER RECORDS, ENTER 3
TO EXIT EMPLOYEE ACCESS, ENTER 4
ENTER YOUR CHOICE: 1
*****
DISPLAYING ROOM PRICES.....
```

=====PRICES PER NIGHT (INR)=====

SNO	FLOORS	PRICE
1	FIRST FLOOR	1000
2	SECOND FLOOR	1600
3	THIRD FLOOR	2200

ENTER 1 TO MODIFY FIRST FLOOR
ENTER 2 TO MODIFY SECOND FLOOR
ENTER 3 TO MODIFY THIRD FLOOR

ENTER FLOOR NUMBER: 2
ENTER NEW PRICE: 1500

=====PRICES CHANGED=====

SNO	FLOORS	PRICE
2	SECOND FLOOR	1500

CHANGE ANOTHER PRICE?(Y/N) y

ENTER 1 TO MODIFY FIRST FLOOR
ENTER 2 TO MODIFY SECOND FLOOR
ENTER 3 TO MODIFY THIRD FLOOR

ENTER FLOOR NUMBER: 1
ENTER NEW PRICE: 1200

=====PRICES CHANGED=====

SNO	FLOORS	PRICE
1	FIRST FLOOR	1200

CHANGE ANOTHER PRICE?(Y/N) n

GOING BACK TO EMPLOYEE ACCESS....

5.2) MODIFYING FOOD PRICES

```
=====
TO MODIFY ROOM PRICES, ENTER 1
TO MODIFY FOOD PRICES, ENTER 2
TO DISPLAY CUSTOMER RECORDS,ENTER 3
TO EXIT EMPLOYEE ACCESS,ENTER 4
ENTER YOUR CHOICE: 2
*****
DISPLAYING FOOD PRICES.....
```

```
-----
=====--HOTEL MENU=====
```

SNO	MENU	PRICE
1	DESSERT	130
2	BEVERAGES	40
3	BREAKFAST	150
4	LUNCH	200
5	DINNER	220

```
-----
ENTER 1 TO MODIFY DESSERT PRICES
ENTER 2 TO MODIFY BEVERAGE PRICES
ENTER 3 TO MODIFY BREAKFAST PRICES
ENTER 4 TO MODIFY LUNCH PRICES
ENTER 5 TO MODIFY DINNER PRICES
-----
```

```
ENTER FOOD CHOICE: 1
ENTER NEW PRICE: 120
-----
```

```
=====--PRICES CHANGED=====
```

SNO	MENU	PRICE
1	DESSERT	120

CHANGE ANOTHER PRICE?(Y/N) y

ENTER 1 TO MODIFY DESSERT PRICES
ENTER 2 TO MODIFY BEVERAGE PRICES
ENTER 3 TO MODIFY BREAKFAST PRICES
ENTER 4 TO MODIFY LUNCH PRICES
ENTER 5 TO MODIFY DINNER PRICES

ENTER FOOD CHOICE: 2
ENTER NEW PRICE: 50

=====PRICES CHANGED=====

SNO	MENU	PRICE
2	BEVERAGES	50

CHANGE ANOTHER PRICE?(Y/N) y

ENTER 1 TO MODIFY DESSERT PRICES
ENTER 2 TO MODIFY BEVERAGE PRICES
ENTER 3 TO MODIFY BREAKFAST PRICES
ENTER 4 TO MODIFY LUNCH PRICES
ENTER 5 TO MODIFY DINNER PRICES

ENTER FOOD CHOICE: 4
ENTER NEW PRICE: 150

=====PRICES CHANGED=====

SNO	MENU	PRICE
4	LUNCH	150

CHANGE ANOTHER PRICE?(Y/N) n
GOING BACK TO EMPLOYEE ACCESS....
=====

5.3) DISPLAYING CUSTOMER RECORDS

TO MODIFY ROOM PRICES, ENTER 1
TO MODIFY FOOD PRICES, ENTER 2
TO DISPLAY CUSTOMER RECORDS, ENTER 3
TO EXIT EMPLOYEE ACCESS, ENTER 4
ENTER YOUR CHOICE: 3

=====DISPLAYING CUSTOMER RECORDS=====

ROOM NO	C_NAME	CHECK_IN	CHECK_OUT
101	JOHN PARKER	2020-12-15	2020-12-23
107	SARAH MAY	2020-12-21	2020-12-27
108	OLIVER FRANK	2020-12-24	2021-01-03
202	LEO DAN	2020-12-13	2020-12-20
203	IBRAHIM ALI	2020-12-11	2020-12-18
207	JENIFER LEE	2020-12-21	2021-01-02
302	ROBERT WOOD	2020-12-11	2020-12-21
305	HANNAH PAUL	2020-12-16	2020-12-20
307	JAMES THOMAS	2020-12-17	2020-12-31
308	RICHARD NOAH	2020-12-12	2021-01-04

TO MODIFY ROOM PRICES, ENTER 1
TO MODIFY FOOD PRICES, ENTER 2
TO DISPLAY CUSTOMER RECORDS, ENTER 3
TO EXIT EMPLOYEE ACCESS, ENTER 4
ENTER YOUR CHOICE: 4

EXITING EMPLOYEE ACCESS...

REFERENCE MANUAL

- ❖ Computer Science with Python by Preeti Arora
- ❖ <https://www.python.org/doc/versions/>
- ❖ Computer Science with Python by Sumita Arora
- ❖ <https://www.w3schools.com/python>
- ❖ <https://sultan-chand.com/ws/python12/#p=1>

CONCLUSION

I am happy to present this project on '*HOTEL MANAGEMENT*'. Our team has been successful in implementing the Project in Python with connectivity to SQL so as to enable permanent storage of data.

Working on the Project has helped me explore the different scopes of Python. It has enabled me to analyse

I hope that my Project is of help to some extent. I have had a great opportunity to learn and research on this topic.

LIMITATIONS

- ❖ The project will not work if the input data does not match the actual data types.
- ❖ Graphics could be added to the program code.
- ❖ The code could be more condensed.