Name of students: Rohan J S, Adarsh Raman

About python

An object orienrted programming languagethats simple. It was developed by Guido Van rossum.

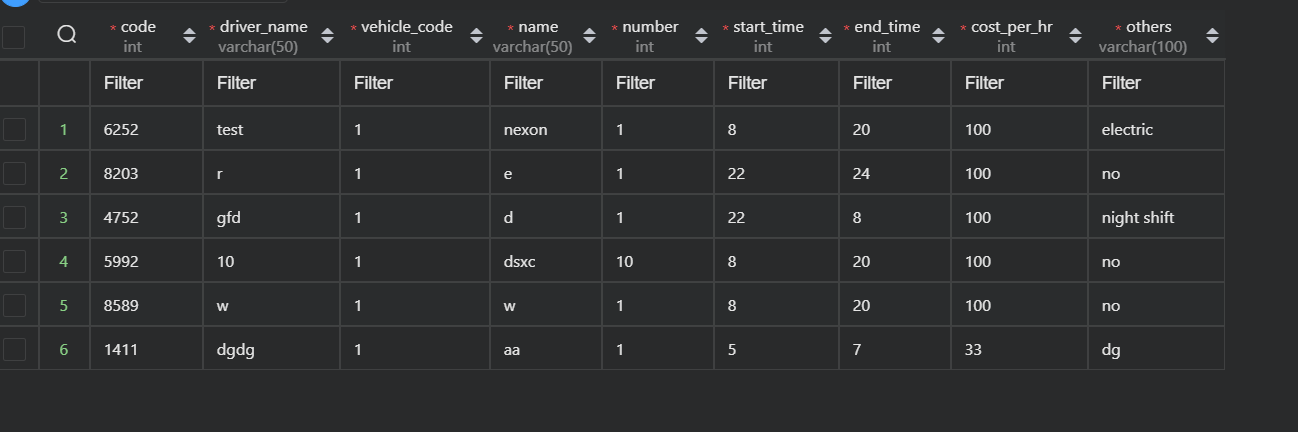
Project intro  
Our project is a Vehicle booking system where you can register as a service provider or a customer.  
As a service provider you will be required to give certain necessary details that will be saved on the database. As a customer you are required to enter the auto generated service provider code and timings of rental before the booking is processed.

1. service providers: service providers can use the application as a platform to accept bookings, registering their vehicles to be available for rent on the application. (Assumed to be drivers who believe in bigger profit margins using organized platforms or companies supplying drivers and vehicles).
2. customers: The customers will be shown the available vehicles for booking. They are to pick a convenient time slot for the booking depending on availability.

Modules involved in the project: , datetime, sqllite3, random, tabulate

Time required for completion of the work: 20 hrs

Table Structure



Source Code

|  |
| --- |
| import sqlite3 |
|  | from time\_module import timer |
|  | base = sqlite3.connect("school.db") |
|  | c = base.cursor() |
|  | from random import randint |
|  | from datetime import datetime |
|  | from tabulate import tabulate |
|  |  |
|  | # c.execute("drop table service\_providers") |
|  | # c.execute("""create table service\_providers( |
|  | # code int primary key, |
|  | # driver\_name varchar(50), |
|  | # vehicle\_code int, |
|  | # name varchar(50) , |
|  | # number int, |
|  | # start\_time int, |
|  | # end\_time int, |
|  | # cost\_per\_hr int, |
|  | # others varchar(100) |
|  |  |
|  |  |
|  |  |
|  | # ) |
|  | # """) |
|  | # c.execute("drop table customers\_log") |
|  | # c.execute("""create table customers\_log( |
|  | # date datetime, |
|  | # customer\_name varchar(100) primary key, |
|  | # adress varchar(100), |
|  | # code int, |
|  | # vehicle\_code int, |
|  | # start\_time int, |
|  | # end\_time int |
|  |  |
|  | # ) |
|  | # """) |
|  |  |
|  |  |
|  |  |
|  | def service\_providers(): |
|  | print("Welcome to.............\ |
|  | please fill in the following few details to sign up as a service provider\ |
|  | please note:\ |
|  | overnight service isnt supported\ |
|  | after every trip, a buffer period of 1 hour will be alloted" |
|  | ) |
|  | # c.execute('Delete from service\_providers') |
|  | # l = [] |
|  | # f1 = open("service\_providers.dat" , 'ab') |
|  | # f2 = open("service\_providers.dat" , 'rb') |
|  | # try: |
|  | # while 1: |
|  | # d = pickle.load(f2) # to load all alloted codes to a list to prevent duplication |
|  | # l += [d] |
|  | # except: |
|  | # f2.close() |
|  | #to check if code aldready exists |
|  | code\_name\_1 = input("enter name of organization/driver: ") |
|  | code\_1 = randint(1000 , 9999) |
|  | data = c.execute("select code from service\_providers") |
|  |  |
|  | l\_temp = set( [ x[0] for x in data.fetchall() ] ) |
|  | print(l\_temp) |
|  |  |
|  | while code\_1 in l\_temp: #checks if the code aldready exists (to be verified) |
|  | code\_1 = randint(1000 , 9999) |
|  |  |
|  | # while {code\_1 , 1} in l: #dict with key as code and value as one to check if it is repeated |
|  | # code\_1 = randint(1000 , 9999) |
|  | # d = {code\_1: 1} |
|  | # pickle.dump(d , f1) |
|  | # f1.close() |
|  |  |
|  | types = int(input("enter number of car models: ")) |
|  | vehicle\_code\_1 = 0 |
|  | for x in range(types): |
|  | vehicle\_code\_1 += 1 |
|  |  |
|  | name\_1 = input("enter car name: ") |
|  | number\_1 = int(input("enter number of units: ")) |
|  | start\_time\_1 = int(input("enter start time (hour in integer format in 24 hour clock): ")) |
|  | end\_time\_1 = int(input("enter end time (hour in integer format in 24 hour clock): ")) |
|  | c1 = start\_time\_1 not in range(0,24) or end\_time\_1 not in range(0 , 24) |
|  | c2 = start\_time\_1 > end\_time\_1 |
|  | while c1 or c2: |
|  | if c1: |
|  | print("sorry that the time given wasnt valid") |
|  | else: |
|  | print("overnight shifts not allowed") |
|  | start\_time\_1 = int(input("enter start time (hour in integer format in 24 hour clock): ")) |
|  | end\_time\_1 = int(input("enter end time (hour in integer format in 24 hour clock): ")) |
|  | c1 = start\_time\_1 not in range(0,24) or end\_time\_1 not in range(0 , 24) |
|  | c2 = start\_time\_1 > end\_time\_1 |
|  |  |
|  | # while start\_time\_1 > end\_time\_1: |
|  | # print("sorry overnight shifts not allowed") |
|  | # start\_time\_1 = int(input("enter start time (hour in integer format in 24 hour clock): ")) |
|  | # end\_time\_1 = int(input("enter end time (hour in integer format in 24 hour clock): ")) |
|  |  |
|  | others\_1 = input("enter any other information (less than 100 characters): ") |
|  | # available\_1 = "Yes" |
|  | cost\_per\_hr\_1 = int(input("enter cost per hour in rupees: ")) |
|  | c.execute("insert into service\_providers(code , driver\_name ,vehicle\_code, name,number, start\_time ,end\_time , cost\_per\_hr , others) values({}, '{}',{}, '{}' , {} ,{}, {} , {} , '{}')".format(code\_1 , code\_name\_1 ,vehicle\_code\_1, name\_1 ,number\_1, start\_time\_1 , end\_time\_1 , cost\_per\_hr\_1 , others\_1)) #syntax for user input insert into table |
|  | #driver name, name(vehicle name), number(units), time are user input code and vehicle\_code are autogenerated available will depend on number |
|  | print("thank you for registering with us") |
|  | base.commit() |
|  |  |
|  | def customers(): |
|  | print("Greetings customer, please fill in the necessary details to proceed with the booking") |
|  | address = input('enter your address: ') |
|  | customer\_name\_1 = input('please enter your name: ') |
|  | temp\_list = [('code' , 'name' , 'vehicle code' , 'car model' , 'units available' , 'start time' , "end time" , 'cost per hour' , 'other information')] |
|  |  |
|  | #print("code name vehicle code car model units available duration of rental") #display of service provider details |
|  | data = c.execute("select \* from service\_providers") |
|  | temp\_list += data.fetchall() |
|  | print(""" |
|  | 1. pick up asap |
|  | 2. pick up at scheduled time |
|  | """) |
|  | choice\_1 = int(input("enter your choice(1 or 2): ")) |
|  | while choice\_1 not in [1,2]: |
|  | print('sorry that input was not valid') |
|  | choice\_1 = int(input("enter your choice(1 or 2): ")) |
|  |  |
|  | if choice\_1 == 1: |
|  | current\_time = datetime.now().hour |
|  |  |
|  |  |
|  | currentDay = datetime.now().day |
|  | if len(str(currentDay)) == 1: |
|  | currentDay = '0' + str(currentDay) |
|  | else: |
|  | currentDay = str(currentDay) |
|  |  |
|  | currentMonth = datetime.now().month |
|  | if len(str(currentMonth)) == 1: |
|  | currentMonth = '0' + str(currentMonth) |
|  | else: |
|  | currentMonth = str(currentMonth) |
|  |  |
|  | currentYear = datetime.now().year |
|  |  |
|  | entered\_date = str(currentYear) + "-" + str(currentMonth) + "-"+ str(currentDay) #to check availability based on date |
|  | time = int(input("enter number of hours of rental: ")) |
|  | while current\_time + time >= 24: |
|  | print("sorry overnight bookings not available") |
|  | time = int(input("enter number of hours of rental: ")) |
|  |  |
|  | else: |
|  | entered\_date = input("enter booking date (format YYYY-MM-DD): ") |
|  | current\_time = int(input("enter pickup time (in 24 hour clock): ")) |
|  | time = int(input("enter number of hours of rental: ")) #vehicle number to be reduces by one for the specified time period |
|  | while current\_time + time >= 24: |
|  | print("sorry overnight bookings not available") |
|  | time = int(input("enter number of hours of rental: ")) |
|  |  |
|  | con\_1 = int(entered\_date[:4]) < datetime.now().year |
|  | con\_2 = int(entered\_date[5:7]) < datetime.now().month and int(entered\_date[:4]) == datetime.now().year |
|  | con\_3 = int(entered\_date[8:10]) < datetime.now().day and int(entered\_date[5:7]) == datetime.now().month and int(entered\_date[:4]) == datetime.now().year |
|  | con\_4 = current\_time <= datetime.now().hour and int(entered\_date[8:10]) == datetime.now().day and int(entered\_date[5:7]) == datetime.now().month and int(entered\_date[:4]) == datetime.now().year |
|  | print(current\_time , datetime.now().hour , con\_4 ) |
|  | '''con 1 con 2 con 3 and con 4 are conditions to ensure that the date enteres isnt from the past''' |
|  |  |
|  | while current\_time not in range(0 , 24) or con\_1 or con\_2 or con\_3 or con\_4: |
|  | print('sorry that input was not valid') |
|  | entered\_date = input("enter booking data (format YYYY-MM-DD): ") |
|  | current\_time = int(input("enter pickup time (in 24 hour clock): ")) |
|  | con\_1 = int(entered\_date[:4]) < datetime.now().year |
|  | con\_2 = int(entered\_date[5:7]) < datetime.now().month and int(entered\_date[:4]) == datetime.now().year |
|  | con\_3 = int(entered\_date[8:10]) < datetime.now().day and int(entered\_date[5:7]) == datetime.now().month and int(entered\_date[:4]) == datetime.now().year |
|  | con\_4 = current\_time <= datetime.now().hour and int(entered\_date[8:10]) == datetime.now().day and int(entered\_date[5:7]) == datetime.now().month and int(entered\_date[:4]) == datetime.now().year |
|  |  |
|  |  |
|  | for x in range(1 , len(temp\_list)): |
|  | if current\_time not in range(temp\_list[x][5] , temp\_list[x][6]): |
|  | a,b,d,e,f,g,h,j,k= temp\_list[x] #unpack and repack |
|  | f = 0 |
|  | temp\_list[x] = (a,b,d,e,f,g,h,j,k) |
|  | """to compare with customers log table and reduce the units for the cars that are currently being used""" |
|  | # data = c.execute("select code , vehicle\_code , start\_time , end\_time from customers\_log") |
|  | data = c.execute("select start\_time , end\_time , date from customers\_log where code = {} and vehicle\_code = {}".format(temp\_list[x][0] , temp\_list[x][2])) |
|  | for y in data.fetchall(): |
|  | c1 = current\_time in range(y[0] , y[1] + 1) and entered\_date == y[2] |
|  | c2 = (current\_time + time) in range(y[0] , y[1] + 1) and entered\_date == y[2] |
|  | c3 = current\_time <= y[0] <= y[1] + 1 <= current\_time + time and entered\_date == y[2] |
|  | print(c1 , c2 , c3) |
|  | print(current\_time , current\_time + time , entered\_date) |
|  | if c1 or c2 or c3: |
|  | a,b,d,e,f,g,h,j,k= temp\_list[x] #unpack and repack |
|  |  |
|  | if f != 0: |
|  | f -= 1 |
|  | temp\_list[x] = (a,b,d,e,f,g,h,j,k) |
|  |  |
|  |  |
|  |  |
|  | print(tabulate(temp\_list)) |
|  | # for x in data.fetchall(): |
|  | # for y in x: |
|  | # print(y , end = " ") |
|  | # print() |
|  |  |
|  | code = int(input("enter preffered service provider code from the table (0 to abort booking): ")) #user to input one of the service provider codes |
|  | if code == 0: |
|  | return None |
|  | data = c.execute("select code from service\_providers") |
|  | li = [] |
|  | for x in data.fetchall(): |
|  |  |
|  | li += [x[0]] |
|  | print(li) |
|  | while code not in li: |
|  | print("code doesnt exist") |
|  | code = int(input("enter preffered service provider code from the table (0 to abort booking): ")) |
|  | if code == 0: |
|  | return None |
|  |  |
|  |  |
|  | print(".....") |
|  | vehicle\_code = int(input("enter vehicle code: ")) |
|  |  |
|  | data = c.execute("SELECT code , vehicle\_code FROM service\_providers") |
|  | temp\_l = [] |
|  | for x in c.fetchall(): |
|  | temp\_l += [x] |
|  | while (code , vehicle\_code) not in temp\_l: |
|  | print("sorry that vehicle code doesnt exist please try again") |
|  | vehicle\_code = int(input("enter vehicle code: ")) |
|  |  |
|  |  |
|  |  |
|  | # data = c.execute("select code , vehicle\_code , number from service\_providers") # to check if vehicle is available |
|  | data = temp\_list[1::] |
|  | for x in data: |
|  | if x[0] == code and x[2] == vehicle\_code: |
|  |  |
|  | if x[4] > 0: |
|  | # time = int(input("enter number of hours of rental: ")) #vehicle number to be reduces by one for the specified time period |
|  | # if current\_time + time > 24: |
|  | # print("sorry overnight bookings not available") |
|  | # time = int(input("enter number of hours of rental: ")) |
|  |  |
|  | # current\_time = datetime.now().hour #current hour is stored |
|  |  |
|  | # data = c.execute("select \* from service\_providers") |
|  |  |
|  | l = [] |
|  | # for i in data.fetchall(): |
|  | # print(i) |
|  | # l += [i] |
|  | # data = l |
|  | data = temp\_list[1::] # to leave out the headers |
|  |  |
|  | for y in range(len(data)): |
|  |  |
|  | if (code , vehicle\_code) == (data[y][0] , data[y][2]): |
|  | units = data[y][4] |
|  | print(units) |
|  | def reduce(code , vehicle\_code, units , start\_time , end\_time): |
|  | # print(current\_time , time) |
|  | if units == 0 or not(start\_time <= current\_time < current\_time + time <= end\_time ): #to check if time is within valid time limit |
|  | if units == 0: |
|  | print("unsuccesful booking no units available") |
|  | else: |
|  | print("sorry units not available for that time slot") |
|  | return None |
|  |  |
|  | else: |
|  |  |
|  | print('booking succesful! Driver will pick you up at the provided adress at the requested time') |
|  | c.execute("insert into customers\_log(date , customer\_name ,adress, code ,vehicle\_code, start\_time ,end\_time ) values('{}', '{}','{}', {} , {} ,{}, {})".format(entered\_date ,customer\_name\_1, address ,code,vehicle\_code, current\_time , current\_time + 1 + time )) #syntax for user input insert into table |
|  | base.commit() |
|  | return True |
|  | reduce(data[y][0] , data[y][2] , units , data[y][5] , data[y][6]) |
|  |  |
|  |  |
|  |  |
|  | else: |
|  | print("booking unsuccesful no units available") |
|  | customers() |
|  | #recursive call if units unavailable |
|  |  |
|  |  |
|  | # def increase\_units(code , vehicle\_code , units , time): |
|  | # data = c.execute("select \* from service\_providers") |
|  | # l = [] |
|  | # for i in data.fetchall(): |
|  | # print(i) |
|  | # l += [i] |
|  | # data = l |
|  |  |
|  | # for y in range(len(data)): |
|  |  |
|  | # if (code , vehicle\_code) == (data[y][0] , data[y][2]): |
|  | # units += 1 |
|  | # data[y] = (data[y][0] , data[y][1] , data[y][2] , data[y][3] ,units ,data[y][5] , data[y][6]) |
|  |  |
|  | # c.execute('Delete from service\_providers') |
|  | # for z in data: |
|  | # code\_1 , code\_name\_1 ,vehicle\_code\_1, name\_1 ,number\_1, time\_1 , available\_1 = z |
|  |  |
|  | # c.execute("insert into service\_providers(code , driver\_name ,vehicle\_code, name,number, time , available) values({}, '{}',{}, '{}' , {} , {},'{}')".format(code\_1 , code\_name\_1 ,vehicle\_code\_1, name\_1 ,number\_1, time\_1 , available\_1)) |
|  | # base.commit() |
|  |  |
|  |  |
|  |  |
|  | # the\_func = Thread(target = customers.func) |
|  | # the\_func.start() |
|  |  |
|  |  |
|  | #main |
|  |  |
|  | choice = input('enter c for customer and s for service provider: ') |
|  | while choice not in ['c' , 's']: |
|  | print("sorry that was an invalid input please input ('c' or 's') ") |
|  | choice = input('enter c for customer and s for service provider: ') |
|  |  |
|  |  |
|  |  |
|  | if choice == 's': |
|  | try: #exception handler to catch wrong input data type |
|  | service\_providers() |
|  | data = c.execute("select \* from service\_providers") |
|  | for x in data.fetchall(): |
|  | print(x) |
|  | except: |
|  | print("sorry an error was raised please adhere to the input instructions") |
|  | print("redirecting.....") |
|  | service\_providers() |
|  |  |
|  | elif choice == 'c': |
|  | # try: #exception handler to catch wrong input data type |
|  | customers() |
|  | # except: |
|  | # print("sorry an error was raised please adhere to the input instructions") |
|  | # print("redirecting.....") |
|  | # customers() |
|  |  |
|  |  |
|  | #function that checks if times in start and end time |
|  | # 3 parameters: start time , end time, duration of rental |
|  | # checks if current time , and current time + duration of rental is within start and end time |
|  |  |
|  |  |