

DBMS MINI PROJECT

CAR RENTAL MANAGEMENT SYSTEM

MEMBERS

RAGHAVAN RAMESH-205002071

VARSHA H – 205002120

SNEHA R- 205002093

DBMS MINI PROJECT – CAR RENTAL MANAGEMENT SYSTEM

Design a Database project for Car Rental System. Agents can register and provide all details about the cars they wish to provide for rent with tariff and terms and conditions, Customers can register and choose the car they want to rent. Agencies can Update, Add, Delete cars in the Database. Customers can view the variety of cars available for in rent in their Budget.

Real time Application:

In real time, we store the details of the cars and their associated agents

Requirement Analysis:

The Customer gives his / her budget and we display the Car details within their budget and the agents of the cars

DATABASE SCHEMA:

Agent Table:

Agent_id	Agent_name	Agent_phone_no	<u>Agent_email</u>	A_password
----------	------------	----------------	--------------------	------------

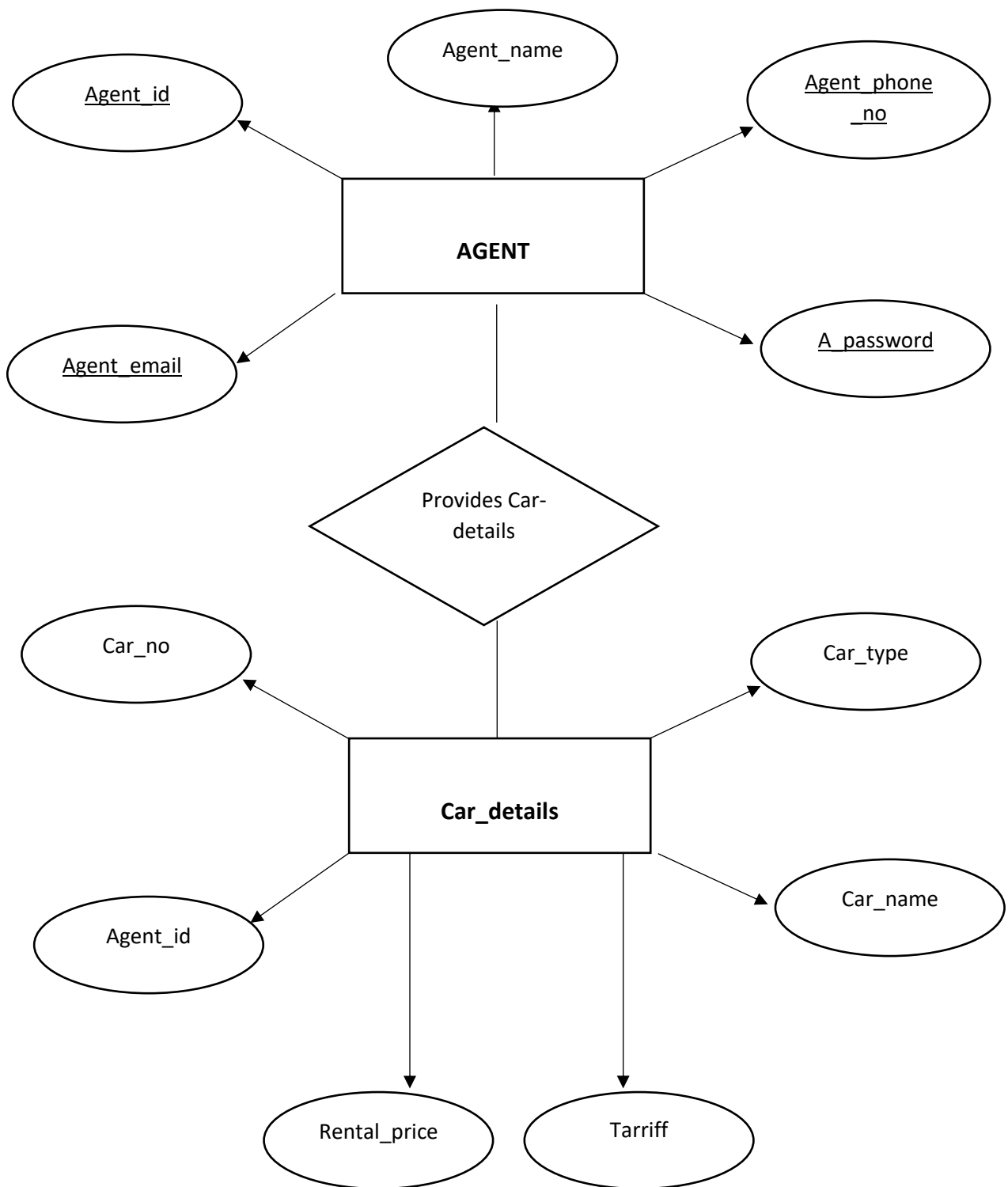
Car_Details Table:

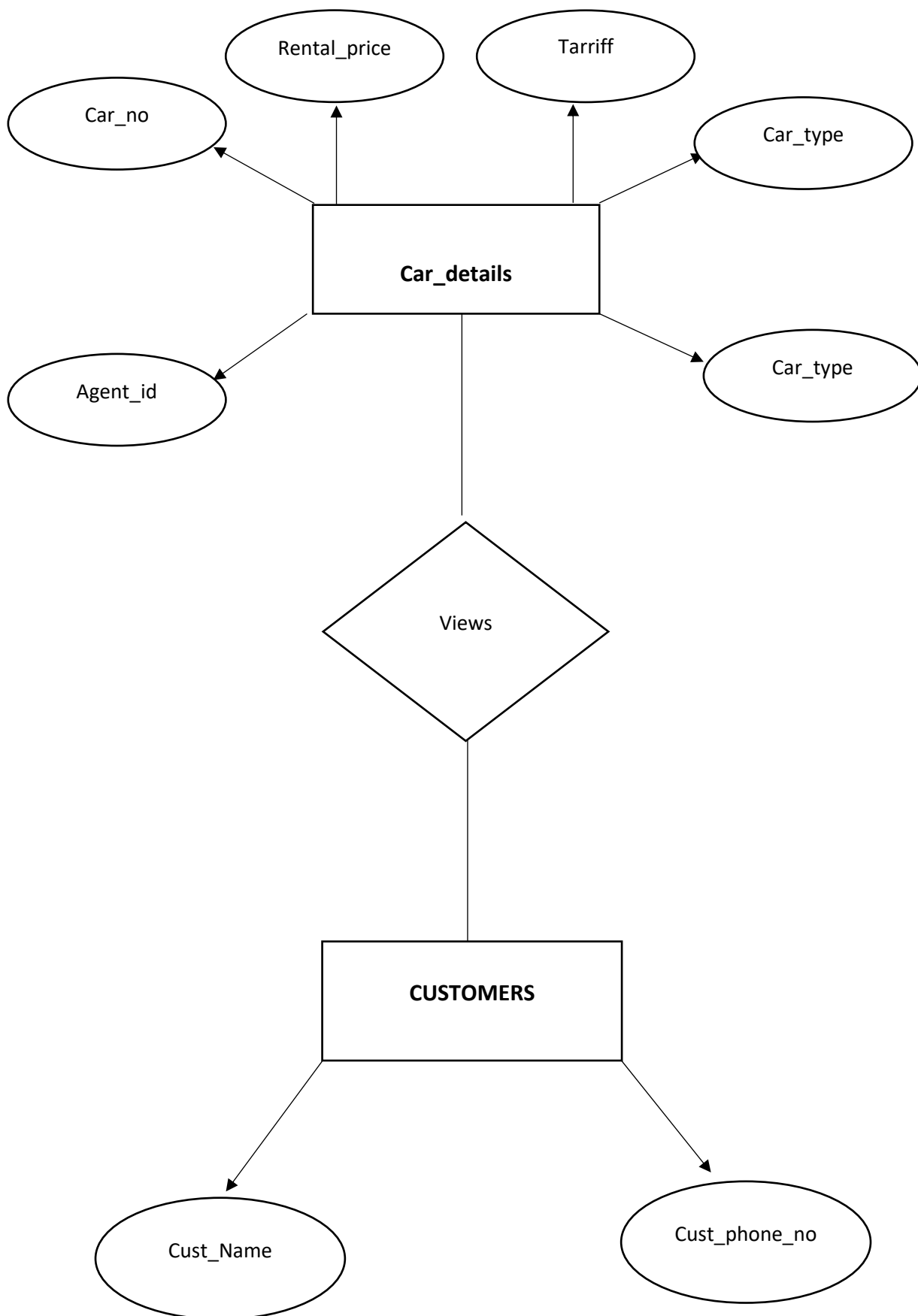
<u>Car_no</u>	Car_type	Agent_id	Rental_price	Tariff	Car_name
---------------	----------	----------	--------------	--------	----------

Customers Table:

Cust_Name	<u>Cust_Phone_no</u>	Exp_price
-----------	----------------------	-----------

E-R Diagrams





Normalisation:

Un normalised Table 1:

Agents:

We are creating a table Agents which keeps track of all the agent details and their associated cars with Agent_email as the primary key.

Agent_id	Agent_name	Agent_phone_no	<u>Agent_email</u>	Agent_Password	Car_no	Car_type	Tariff
----------	------------	----------------	--------------------	----------------	--------	----------	--------

Normalisation:

Agent_id	Agent_name	Agent_phone_no	<u>Agent_email</u>	Agent_Password	Car_no	Car_type	Tariff
<u>0_varsha@127</u>	Varsha	5678291021	varsha@127	bts	566	XUV	8000
0_sneha@123	Sneha	4567890211	sneha@123	mondler	6718	Sedon	6000
<u>0_raghav@536</u>	Raghavan	3456789032	raghav@536	rts	528	Hatchback	4000

- All the attributes in this table is Single Valued, so it is 1 NF.

Agent_id	Agent_name	Agent_phone_no	<u>Agent_email</u>	Agent_Password	Car_no	Car_type	Price
<u>0_varsha@127</u>	Varsha	5678291021	varsha@127	bts	566	XUV	8000
0_sneha@123	Sneha	4567890211	sneha@123	mondler	6718	Sedon	6000
<u>0_raghav@536</u>	Raghavan	3456789032	raghav@536	rts	528	Hatchback	4000

- The attribute Agent_id is dependent on Agent_email which is the primary key of the table
- The attribute Car_no depends on Non- primary derived attribute, Agent_id which depends on the primary key
- Since Car_no is dependent on Agent_id it is transitive dependencey
- So we split the table into two different tables with Car no as the primary key of the second table

Table 1: Agents Table:

Agent_id	Agent_name	Agent_phone_no	<u>Agent_email</u>	Agent_Password
<u>0_varsha@127</u>	Varsha	5678291021	varsha@127	bts
0_sneha@123	Sneha	4567890211	sneha@123	mondler
<u>0_raghav@536</u>	Raghavan	3456789032	raghav@536	rts

Table 2: Car_details Table:

Car_no	Car_type	Price
566	XUV	8000
6718	Sedon	6000
528	Hatchback	4000

Table 2: Car_details Table:

Car_no	Car_type	Price
566	XUV	8000
6718	Sedon	6000
528	Hatchback	4000

- Addition of Attributes in the car_details Table due to the requirements

Table 2: Car_details Table:

Car_no	Car_type	Rental_Price	Tarriff	Agent_id	Car_name
566	SUV	8000	2000	<u>0_varsha@127</u>	XUV 500
6718	Sedon	6000	1500	<u>0_sneha@123</u>	<u>Honda Amaze</u>
528	Hatchback	4000	1200	<u>0_raghav@536</u>	<u>Hyundai i10</u>

Code:

Main.py

```
import tkinter as tk
import tkinter.font as tkFont

class App:
    def __init__(self, root):
        #setting title
        root.title("SVR CAR RENTAL MEDIATION SERVICE")
        #setting window size
        width=600
        height=500
        screenwidth = root.winfo_screenwidth()
        screenheight = root.winfo_screenheight()
        alignstr = '%dx%d+%d+%d' % (width, height, (screenwidth - width) /
2, (screenheight - height) / 2)
        root.geometry(alignstr)
        root.resizable(width=False, height=False)
        root.config(bg="yellow")

        GButton_842=tk.Button(root)
        GButton_842["bg"] = "#070303"
        ft = tkFont.Font(family='Times',size=10)
        GButton_842["font"] = ft
        GButton_842["fg"] = "#f11b1b"
        GButton_842["justify"] = "center"
        GButton_842["text"] = "CUSTOMER LOGIN"
        GButton_842.place(x=190,y=140,width=200,height=25)
        GButton_842["command"] = customer_login

        GButton_41=tk.Button(root)
        GButton_41["bg"] = "#0e0d0d"
        ft = tkFont.Font(family='Times',size=10)
        GButton_41["font"] = ft
        GButton_41["fg"] = "#ed1818"
        GButton_41["justify"] = "center"
        GButton_41["text"] = "EXISTING AGENT LOGIN"
        GButton_41.place(x=190,y=230,width=200,height=25)
        GButton_41["command"] = agent_login

        GButton_972=tk.Button(root)
        GButton_972["bg"] = "#000000"
        ft = tkFont.Font(family='Times',size=10)
        GButton_972["font"] = ft
        GButton_972["fg"] = "#ef1e1e"
        GButton_972["justify"] = "center"
        GButton_972["text"] = "NEW AGENT REGISTRATION"
        GButton_972.place(x=190,y=320,width=200,height=25)
        GButton_972["command"] = agent_registration

        GLabel_945=tk.Label(root)
        GLabel_945["bg"] = "#040404"
        ft = tkFont.Font(family='Times',size=10)
        GLabel_945["font"] = ft
        GLabel_945["fg"] = "#e31717"
        GLabel_945["justify"] = "center"
        GLabel_945["text"] = "SVR CAR RENTAL MEDIATION SERVICE"
        GLabel_945.place(x=140,y=50,width=300,height=25)
```



```

def customer_login():
    import Customer_View
    Customer_View.root = tk.Toplevel(root)
    Customer_View.root.resizable(True, True)
    Customer_View.app_1 = Customer_View.App(Customer_View.root)
    Customer_View.root.config(bg="yellow")
    Customer_View.root.mainloop()

def agent_login():
    import Agent_Login
    Agent_Login.root = tk.Toplevel(root)
    Agent_Login.root.resizable(True, True)
    Agent_Login.app_2 = Agent_Login.App(Agent_Login.root)
    Agent_Login.root.config(bg="yellow")
    Agent_Login.root.mainloop()

def agent_registration():
    import Agent_Registration
    Agent_Registration.root = tk.Toplevel(root)
    Agent_Registration.root.resizable(True, True)
    Agent_Registration.app_3 =
Agent_Registration.App(Agent_Registration.root)
    Agent_Registration.root.config(bg="yellow")
    Agent_Registration.root.mainloop()

root = tk.Tk()
app = App(root)
root.mainloop()

```

Customer_View.py

```

import tkinter as tk
import tkinter.font as tkFont
import mysql.connector
from tkinter import *
from tkinter import messagebox

class App:
    GLineEdit_525 = None
    GLineEdit_438 = None
    GLineEdit_944 = None
    GListBox_320 = None

    def __init__(self, root):
        #setting title
        root.title("CUSTOMER VIEW")
        #setting window size
        width=100
        height=40
        screenwidth = root.winfo_screenwidth()
        screenheight = root.winfo_screenheight()
        alignstr = '%dx%d+%d+%d' % (width, height, (screenwidth - width) /
2, (screenheight - height) / 2)
        root.geometry(alignstr)

```

```

root.resizable(width=False, height=False)

GLabel_784=tk.Label(root)
GLabel_784["bg"] = "#100f0f"
ft = tkFont.Font(family='Times',size=10)
GLabel_784["font"] = ft
GLabel_784["fg"] = "#fafa1a"
GLabel_784["justify"] = "center"
GLabel_784["text"] = "Name"
GLabel_784.place(x=40,y=50,width=70,height=25)

GLabel_507=tk.Label(root)
GLabel_507["bg"] = "#150707"
ft = tkFont.Font(family='Times',size=10)
GLabel_507["font"] = ft
GLabel_507["fg"] = "#e21c1c"
GLabel_507["justify"] = "center"
GLabel_507["text"] = "Phone Number"
GLabel_507.place(x=40,y=100,width=70,height=25)

App.GLineEdit_525=tk.Entry(root)
App.GLineEdit_525["borderwidth"] = "1px"
ft = tkFont.Font(family='Times',size=10)
App.GLineEdit_525["font"] = ft
App.GLineEdit_525["fg"] = "#333333"
App.GLineEdit_525["justify"] = "center"
App.GLineEdit_525["text"] = ""
App.GLineEdit_525.place(x=170,y=50,width=200,height=25)

App.GLineEdit_438=tk.Entry(root)
App.GLineEdit_438["borderwidth"] = "1px"
ft = tkFont.Font(family='Times',size=10)
App.GLineEdit_438["font"] = ft
App.GLineEdit_438["fg"] = "#333333"
App.GLineEdit_438["justify"] = "center"
App.GLineEdit_438["text"] = ""
App.GLineEdit_438.place(x=170,y=100,width=200,height=25)

App.GListBox_320=tk.Listbox(root)
App.GListBox_320["bg"] = "#f8f3f3"
App.GListBox_320["borderwidth"] = "1px"
ft = tkFont.Font(family='Times',size=10)
App.GListBox_320["font"] = ft
App.GListBox_320["fg"] = "#333333"
App.GListBox_320["justify"] = "center"
App.GListBox_320.place(x=20,y=200,width=602,height=336)
App.GListBox_320["exportselection"] = "0"
App.GListBox_320["selectmode"] = "browse"
App.GListBox_320["setgrid"] = "True"

GLabel_32=tk.Label(root)
GLabel_32["bg"] = "#232020"
ft = tkFont.Font(family='Times',size=10)
GLabel_32["font"] = ft
GLabel_32["fg"] = "#de1414"
GLabel_32["justify"] = "center"
GLabel_32["text"] = "Maximum Expected Price"
GLabel_32.place(x=10,y=150,width=142,height=30)

App.GLineEdit_944=tk.Entry(root)
App.GLineEdit_944["borderwidth"] = "1px"

```

```

ft = tkFont.Font(family='Times',size=10)
App.GLineEdit_944["font"] = ft
App.GLineEdit_944["fg"] = "#333333"
App.GLineEdit_944["justify"] = "center"
App.GLineEdit_944["text"] = ""
App.GLineEdit_944.place(x=170,y=150,width=200,height=25)

GButton_940=tk.Button(root)
GButton_940["bg"] = "#1b1717"
ft = tkFont.Font(family='Times',size=10)
GButton_940["font"] = ft
GButton_940["fg"] = "#ed1b1b"
GButton_940["justify"] = "center"
GButton_940["text"] = "Submit"
GButton_940.place(x=490,y=60,width=70,height=25)
GButton_940["command"] = submit

GButton_417=tk.Button(root)
GButton_417["bg"] = "#242121"
ft = tkFont.Font(family='Times',size=10)
GButton_417["font"] = ft
GButton_417["fg"] = "#e71e1e"
GButton_417["justify"] = "center"
GButton_417["text"] = "Exit"
GButton_417.place(x=490,y=120,width=70,height=25)
GButton_417["command"] = lambda: root.destroy()

def submit():
    conn = mysql.connector.connect(host="localhost", user="root",
password="Rr2163?!", database="car_rentals")
    cur_1 = conn.cursor()
    cur_2 = conn.cursor()

    name = App.GLineEdit_525.get()
    phone_no = App.GLineEdit_438.get()
    max_price = App.GLineEdit_944.get()

    cur_2.execute("insert into customers
values('{}','{}','{}').format(name,phone_no,max_price))

    cur_1.execute("select C.car_no,
C.car_type,C.rental_price,C.tariff,C.car_name,A.agent_name,A.agent_phone_no
,A.agent_email from car_details C natural join agents A where
C.rental_price + C.tariff <= '{}'.format(max_price))
    m = cur_1.fetchall()

    App.GListBox_320.insert(END, "
Car No      Car Type      RentalPrice
Tariff      Car Name      AgentName      AgentPhoneNum      AgentEmail ")
    App.GListBox_320.insert(END, "-----")
    App.GListBox_320.insert(END, "-----")
    for i in m:
        App.GListBox_320.insert(END, str(i[0]) + "
" + str(i[2]) + "
" + str(i[3]) + "
" + str(i[4]) + "
" + str(i[5]) + "
" + str(i[6]) + "
" + str(i[7]))

    if cur_1.rowcount == 0:
        messagebox.showinfo("Sorry","No cars exist within your budget!")

    conn.commit()
    conn.close()

```

Agent_Login.py

```
import tkinter as tk
import tkinter.font as tkFont
import mysql.connector
from tkinter import messagebox

class App:
    GLineEdit_897 = None
    GLineEdit_736 = None
    root = None

    def __init__(self, root):
        #setting title
        root.title("EXISTING AGENT LOGIN")
        #setting window size
        width=600
        height=500
        screenwidth = root.winfo_screenwidth()
        screenheight = root.winfo_screenheight()
        alignstr = '%dx%d+%d+%d' % (width, height, (screenwidth - width) /
2, (screenheight - height) / 2)
        root.geometry(alignstr)
        root.resizable(width=False, height=False)

        GLabel_780=tk.Label(root)
        GLabel_780["bg"] = "#171414"
        ft = tkFont.Font(family='Times',size=10)
        GLabel_780["font"] = ft
        GLabel_780["fg"] = "#e01919"
        GLabel_780["justify"] = "center"
        GLabel_780["text"] = "Email ID"
        GLabel_780.place(x=50,y=90,width=70,height=25)

        GLabel_369=tk.Label(root)
        GLabel_369["bg"] = "#180404"
        ft = tkFont.Font(family='Times',size=10)
        GLabel_369["font"] = ft
        GLabel_369["fg"] = "#de1919"
        GLabel_369["justify"] = "center"
        GLabel_369["text"] = "Password"
        GLabel_369.place(x=50,y=160,width=70,height=25)

        GButton_464=tk.Button(root)
        GButton_464["bg"] = "#1b1919"
        ft = tkFont.Font(family='Times',size=10)
        GButton_464["font"] = ft
        GButton_464["fg"] = "#e60f0f"
        GButton_464["justify"] = "center"
        GButton_464["text"] = "Submit"
        GButton_464.place(x=130,y=250,width=70,height=25)
        GButton_464["command"] = log_in

        App.GLineEdit_897=tk.Entry(root)
        App.GLineEdit_897["borderwidth"] = "1px"
```

```

ft = tkFont.Font(family='Times',size=10)
App.GLineEdit_897["font"] = ft
App.GLineEdit_897["fg"] = "#333333"
App.GLineEdit_897["justify"] = "center"
App.GLineEdit_897["text"] = ""
App.GLineEdit_897.place(x=190,y=90,width=200,height=25)

App.GLineEdit_736=tk.Entry(root)
App.GLineEdit_736["borderwidth"] = "1px"
ft = tkFont.Font(family='Times',size=10)
App.GLineEdit_736["font"] = ft
App.GLineEdit_736["fg"] = "#333333"
App.GLineEdit_736["justify"] = "center"
App.GLineEdit_736["text"] = ""
App.GLineEdit_736["show"] = "*"
App.GLineEdit_736.place(x=190,y=160,width=200,height=25)

GButton_829=tk.Button(root)
GButton_829["bg"] = "#131212"
ft = tkFont.Font(family='Times',size=10)
GButton_829["font"] = ft
GButton_829["fg"] = "#e41b1b"
GButton_829["justify"] = "center"
GButton_829["text"] = "Exit"
GButton_829.place(x=340,y=250,width=70,height=25)
GButton_829["command"] = lambda: root.destroy()

def log_in():

    conn =
mysql.connector.connect(host="localhost",user="root",password="Rr2163?!",da
tabase="car_rentals")
    cur = conn.cursor()

    e_mailID = App.GLineEdit_897.get()
    pass_word = App.GLineEdit_736.get()

    cur.execute("SELECT agent_id,A_password from agents where agent_email =
'{}' and A_password = '{}'.format(e_mailID,pass_word))
    c = cur.fetchall()
    if c:
        messagebox.showinfo("Success","Welcome to SVR")
        import Agent_View
        Agent_View.root = tk.Toplevel(App.root)
        Agent_View.root.resizable(True,True)
        Agent_View.app = Agent_View.App(Agent_View.root)
        Agent_View.root.config(bg="yellow")
        Agent_View.root.mainloop()

    else:
        messagebox.showinfo("Error","Invalid Login!")

    conn.commit()
    conn.close()

```

Agent_Registration.py

```

import tkinter as tk
import tkinter.font as tkFont
from tkinter import messagebox

```

```

import mysql.connector

global name, phone_no, email, agentTable, root

class App:
    GLineEdit_528 = None
    GLineEdit_239 = None
    GLineEdit_864 = None
    GLineEdit_972 = None
    def __init__(self, root):
        #setting title
        root.title("undefined")
        #setting window size
        width=600
        height=500
        screenwidth = root.winfo_screenwidth()
        screenheight = root.winfo_screenheight()
        alignstr = '%dx%d+%d+%d' % (width, height, (screenwidth - width) /
2, (screenheight - height) / 2)
        root.geometry(alignstr)
        root.resizable(width=False, height=False)

        GLabel_951=tk.Label(root)
        GLabel_951["bg"] = "#250505"
        ft = tkFont.Font(family='Times',size=10)
        GLabel_951["font"] = ft
        GLabel_951["fg"] = "#f12121"
        GLabel_951["justify"] = "center"
        GLabel_951["text"] = "Name"
        GLabel_951.place(x=60,y=90,width=70,height=25)

        GLabel_512=tk.Label(root)
        GLabel_512["bg"] = "#1e0d0d"
        ft = tkFont.Font(family='Times',size=10)
        GLabel_512["font"] = ft
        GLabel_512["fg"] = "#e32424"
        GLabel_512["justify"] = "center"
        GLabel_512["text"] = "Email ID"
        GLabel_512.place(x=60,y=150,width=70,height=25)

        GLabel_696=tk.Label(root)
        GLabel_696["bg"] = "#170606"
        ft = tkFont.Font(family='Times',size=10)
        GLabel_696["font"] = ft
        GLabel_696["fg"] = "#ec2e2e"
        GLabel_696["justify"] = "center"
        GLabel_696["text"] = "Phone"
        GLabel_696.place(x=60,y=210,width=70,height=25)

        GLabel_56=tk.Label(root)
        GLabel_56["bg"] = "#230d0d"
        ft = tkFont.Font(family='Times',size=10)
        GLabel_56["font"] = ft
        GLabel_56["fg"] = "#f63434"
        GLabel_56["justify"] = "center"
        GLabel_56["text"] = "Desired Password"
        GLabel_56.place(x=40,y=260,width=108,height=42)

        App.GLineEdit_528=tk.Entry(root)

```

```

App.GLineEdit_528["borderwidth"] = "1px"
ft = tkFont.Font(family='Times',size=10)
App.GLineEdit_528["font"] = ft
App.GLineEdit_528["fg"] = "#333333"
App.GLineEdit_528["justify"] = "center"
App.GLineEdit_528["text"] = ""
App.GLineEdit_528.place(x=210,y=90,width=200,height=25)

App.GLineEdit_239=tk.Entry(root)
App.GLineEdit_239["borderwidth"] = "1px"
ft = tkFont.Font(family='Times',size=10)
App.GLineEdit_239["font"] = ft
App.GLineEdit_239["fg"] = "#333333"
App.GLineEdit_239["justify"] = "center"
App.GLineEdit_239["text"] = ""
App.GLineEdit_239.place(x=210,y=150,width=200,height=25)

App.GLineEdit_864=tk.Entry(root)
App.GLineEdit_864["borderwidth"] = "1px"
ft = tkFont.Font(family='Times',size=10)
App.GLineEdit_864["font"] = ft
App.GLineEdit_864["fg"] = "#333333"
App.GLineEdit_864["justify"] = "center"
App.GLineEdit_864["text"] = ""
App.GLineEdit_864.place(x=210,y=210,width=200,height=25)

App.GLineEdit_972=tk.Entry(root)
App.GLineEdit_972["borderwidth"] = "1px"
ft = tkFont.Font(family='Times',size=10)
App.GLineEdit_972["font"] = ft
App.GLineEdit_972["fg"] = "#333333"
App.GLineEdit_972["justify"] = "center"
App.GLineEdit_972["text"] = ""
App.GLineEdit_972.place(x=210,y=270,width=200,height=25)

GButton_8=tk.Button(root)
GButton_8["bg"] = "#210404"
ft = tkFont.Font(family='Times',size=10)
GButton_8["font"] = ft
GButton_8["fg"] = "#e61d1d"
GButton_8["justify"] = "center"
GButton_8["text"] = "Register"
GButton_8.place(x=180,y=380,width=70,height=25)
GButton_8["command"] = add_agent

GButton_833=tk.Button(root)
GButton_833["bg"] = "#120a0a"
ft = tkFont.Font(family='Times',size=10)
GButton_833["font"] = ft
GButton_833["fg"] = "#d21717"
GButton_833["justify"] = "center"
GButton_833["text"] = "Exit"
GButton_833.place(x=320,y=380,width=70,height=25)
GButton_833["command"] = lambda: root.destroy()

```

```

def add_agent():
    conn = mysql.connector.connect(host="localhost", user="root",
password="Rr2163?!", database="car_rentals")
    conn.autocommit = True

```

```

cur = conn.cursor()

global name, phone_no, email, agentTable
name = App.GLineEdit_528.get()
phone_no = App.GLineEdit_864.get()
email = App.GLineEdit_239.get()

ID = "0 " + email
password = App.GLineEdit_972.get()

cur.execute( "insert into agents values
('{}','{}','{}','{}','{}')".format(ID, name, phone_no, email,password))
try:
    conn.commit()
    conn.close()
    messagebox.showinfo("Success", "You are added Successfully!")

except:
    messagebox.showinfo("Error", "Couldn't add data to database!")

```

Agent_View.py

```

import tkinter as tk
import tkinter.font as tkFont

class App:

    root = None
    def __init__(self, root):
        #setting title
        root.title("Welcome Agent")
        #setting window size
        width=600
        height=500
        screenwidth = root.winfo_screenwidth()
        screenheight = root.winfo_screenheight()
        alignstr = '%dx%d+%d+%d' % (width, height, (screenwidth - width) /
2, (screenheight - height) / 2)
        root.geometry(alignstr)
        root.resizable(width=False, height=False)

        GButton_370=tk.Button(root)
        GButton_370["bg"] = "#1b0e0e"
        ft = tkFont.Font(family='Times',size=10)
        GButton_370["font"] = ft
        GButton_370["fg"] = "#f32626"
        GButton_370["justify"] = "center"
        GButton_370["text"] = "View"
        GButton_370.place(x=270,y=170,width=88,height=54)
        GButton_370["command"] = open_view

        GButton_267=tk.Button(root)
        GButton_267["bg"] = "#150c0c"
        ft = tkFont.Font(family='Times',size=10)
        GButton_267["font"] = ft
        GButton_267["fg"] = "#f91f1f"
        GButton_267["justify"] = "center"
        GButton_267["text"] = "Update"

```



```

GButton_267.place(x=270,y=260,width=86,height=50)
GButton_267["command"] = open_update

GButton_99=tk.Button(root)
GButton_99["bg"] = "#1b0a0a"
ft = tkFont.Font(family='Times',size=10)
GButton_99["font"] = ft
GButton_99["fg"] = "#f91b1b"
GButton_99["justify"] = "center"
GButton_99["text"] = "Delete"
GButton_99.place(x=270,y=350,width=86,height=51)
GButton_99["command"] = open_delete

GButton_555=tk.Button(root)
GButton_555["bg"] = "#150909"
ft = tkFont.Font(family='Times',size=10)
GButton_555["font"] = ft
GButton_555["fg"] = "#d31616"
GButton_555["justify"] = "center"
GButton_555["text"] = "Add"
GButton_555.place(x=270,y=110,width=84,height=37)
GButton_555["command"] = open_add

def open_add():
    import Add
    Add.root = tk.Toplevel(App.root)
    Add.app = Add.App(Add.root)
    Add.root.config(bg="yellow")
    Add.root.mainloop()

def open_delete():
    import Delete
    Delete.root = tk.Toplevel(App.root)
    Delete.app = Delete.App(Delete.root)
    Delete.root.config(bg="yellow")
    Delete.root.mainloop()

def open_update():
    import Update
    Update.root = tk.Toplevel(App.root)
    Update.app = Update.App(Update.root)
    Update.root.config(bg="yellow")
    Update.root.mainloop()

def open_view():
    import View
    View.root = tk.Toplevel(App.root)
    View.app = View.App(View.root)
    View.root.config(bg="yellow")
    View.root.mainloop()

```

Update.py

```

import tkinter as tk
import tkinter.font as tkFont
import mysql.connector
from tkinter import messagebox

class App:

```

```

GLineEdit_611 = None
GLineEdit_51 = None
def __init__(self, root):
    # setting title
    root.title("undefined")
    # setting window size
    width = 600
    height = 500
    screenwidth = root.winfo_screenwidth()
    screenheight = root.winfo_screenheight()
    alignstr = '%dx%d+%d+%d' % (width, height, (screenwidth - width) /
2, (screenheight - height) / 2)
    root.geometry(alignstr)
    root.resizable(width=False, height=False)

    GButton_925 = tk.Button(root)
    GButton_925["bg"] = "#1e0f0f"
    ft = tkFont.Font(family='Times', size=10)
    GButton_925["font"] = ft
    GButton_925["fg"] = "#db2929"
    GButton_925["justify"] = "center"
    GButton_925["text"] = "Car Number"
    GButton_925.place(x=30, y=70, width=70, height=25)
    GButton_925["command"] = update_car_number

    GButton_195 = tk.Button(root)
    GButton_195["bg"] = "#0c0202"
    ft = tkFont.Font(family='Times', size=10)
    GButton_195["font"] = ft
    GButton_195["fg"] = "#d21e1e"
    GButton_195["justify"] = "center"
    GButton_195["text"] = "Car Type"
    GButton_195.place(x=140, y=70, width=70, height=25)
    GButton_195["command"] = update_car_type

    GButton_903 = tk.Button(root)
    GButton_903["bg"] = "#1a1414"
    ft = tkFont.Font(family='Times', size=10)
    GButton_903["font"] = ft
    GButton_903["fg"] = "#ef2d2d"
    GButton_903["justify"] = "center"
    GButton_903["text"] = "Rental Price"
    GButton_903.place(x=250, y=70, width=70, height=25)
    GButton_903["command"] = update_car_price

    GButton_267 = tk.Button(root)
    GButton_267["bg"] = "#201414"
    ft = tkFont.Font(family='Times', size=10)
    GButton_267["font"] = ft
    GButton_267["fg"] = "#dc2525"
    GButton_267["justify"] = "center"
    GButton_267["text"] = "Tariff"
    GButton_267.place(x=360, y=70, width=70, height=25)
    GButton_267["command"] = update_car_tariff

    GButton_915 = tk.Button(root)
    GButton_915["bg"] = "#291414"
    ft = tkFont.Font(family='Times', size=10)
    GButton_915["font"] = ft
    GButton_915["fg"] = "#e22222"
    GButton_915["justify"] = "center"

```

```

GButton_915["text"] = "Car Name"
GButton_915.place(x=470, y=70, width=70, height=25)
GButton_915["command"] = update_car_name

GLabel_12 = tk.Label(root)
GLabel_12["bg"] = "#1d1414"
ft = tkFont.Font(family='Times', size=10)
GLabel_12["font"] = ft
GLabel_12["fg"] = "#de2626"
GLabel_12["justify"] = "center"
GLabel_12["text"] = "Enter Car Number"
GLabel_12.place(x=30, y=190, width=112, height=31)

GLabel_567 = tk.Label(root)
GLabel_567["bg"] = "#1a0f0f"
ft = tkFont.Font(family='Times', size=10)
GLabel_567["font"] = ft
GLabel_567["fg"] = "#e42020"
GLabel_567["justify"] = "center"
GLabel_567["text"] = "Enter New Value"
GLabel_567.place(x=40, y=270, width=92, height=30)

App.GLineEdit_611 = tk.Entry(root)
App.GLineEdit_611["borderwidth"] = "1px"
ft = tkFont.Font(family='Times', size=10)
App.GLineEdit_611["font"] = ft
App.GLineEdit_611["fg"] = "#333333"
App.GLineEdit_611["justify"] = "center"
App.GLineEdit_611["text"] = ""
App.GLineEdit_611.place(x=160, y=190, width=200, height=25)

App.GLineEdit_51 = tk.Entry(root)
App.GLineEdit_51["borderwidth"] = "1px"
ft = tkFont.Font(family='Times', size=10)
App.GLineEdit_51["font"] = ft
App.GLineEdit_51["fg"] = "#333333"
App.GLineEdit_51["justify"] = "center"
App.GLineEdit_51["text"] = ""
App.GLineEdit_51.place(x=160, y=270, width=200, height=25)

def update_car_number():

    conn = mysql.connector.connect(host="localhost", user="root",
password="Rr2163?!", database="car_rentals")
    cur = conn.cursor()

    new_car_no = App.GLineEdit_51.get()
    old_car_no = App.GLineEdit_611.get()

    cur.execute("update car_details set car_no = '{}' where car_no =
'{}'.format(new_car_no,old_car_no))
    try:
        conn.commit()
        conn.close()
        messagebox.showinfo("Success","Updated Car Number
Successfully!")
    except:
        messagebox.showinfo("Error","Couldn't Update Car Number!")

```

```

def update_car_type():

    conn = mysql.connector.connect(host="localhost", user="root",
password="Rr2163?!", database="car_rentals")
    cur = conn.cursor()

    new_car_type = App.GLineEdit_51.get()
    old_car_no = App.GLineEdit_611.get()

    cur.execute("update car_details set car_type = '{}' where car_no =
'{}'.format(new_car_type, old_car_no))
    try:
        conn.commit()
        conn.close()
        messagebox.showinfo("Success", "Updated Car Type
Successfully!")
    except:
        messagebox.showinfo("Error", "Couldn't Update Car Type!")


def update_car_price():

    conn = mysql.connector.connect(host="localhost", user="root",
password="Rr2163?!", database="car_rentals")
    cur = conn.cursor()

    new_price = App.GLineEdit_51.get()
    old_car_no = App.GLineEdit_611.get()

    cur.execute("update car_details set rental_price = '{}' where
car_no = '{}'.format(new_price, old_car_no))
    try:
        conn.commit()
        conn.close()
        messagebox.showinfo("Success", "Updated Car Price
Successfully!")
    except:
        messagebox.showinfo("Error", "Couldn't Update Car Price!")


def update_car_tariff():

    conn = mysql.connector.connect(host="localhost", user="root",
password="Rr2163?!", database="car_rentals")
    cur = conn.cursor()

    new_tariff = App.GLineEdit_51.get()
    old_car_no = App.GLineEdit_611.get()

    cur.execute("update car_details set tariff = '{}' where car_no =
'{}'.format(new_tariff, old_car_no))
    try:
        conn.commit()
        conn.close()
        messagebox.showinfo("Success", "Updated Car Tariff

```

```

Successfully!")
    except:
        messagebox.showinfo("Error", "Couldn't Update Car Tariff!")

def update_car_name():

    conn = mysql.connector.connect(host="localhost", user="root",
password="Rr2163?!", database="car_rentals")
    cur = conn.cursor()

    new_car_name = App.GLineEdit_51.get()
    old_car_no = App.GLineEdit_611.get()

    cur.execute("update car_details set car_name = '{}' where car_no =
'{}'.format(new_car_name, old_car_no))
    try:
        conn.commit()
        conn.close()
        messagebox.showinfo("Success", "Updated Car Name
Successfully!")
    except:
        messagebox.showinfo("Error", "Couldn't Update Name Number!")

```

Delete.py

```

import tkinter as tk
import tkinter.font as tkFont
import mysql.connector
from tkinter import messagebox

class App:
    GLineEdit_99 = None

    def __init__(self, root):
        # setting title
        root.title("Delete")
        # setting window size
        width = 600
        height = 500
        screenwidth = root.winfo_screenwidth()
        screenheight = root.winfo_screenheight()
        alignstr = '%dx%d+%d+%d' % (width, height, (screenwidth - width) /
2, (screenheight - height) / 2)
        root.geometry(alignstr)
        root.resizable(width=False, height=False)

        GLabel_743 = tk.Label(root)
        GLabel_743["bg"] = "#191818"
        ft = tkFont.Font(family='Times', size=10)
        GLabel_743["font"] = ft
        GLabel_743["fg"] = "#cc4040"
        GLabel_743["justify"] = "center"
        GLabel_743["text"] = "Car Number "
        GLabel_743.place(x=80, y=110, width=139, height=37)

        App.GLineEdit_99 = tk.Entry(root)

```

```

App.GLineEdit_99["borderwidth"] = "1px"
ft = tkFont.Font(family='Times', size=10)
App.GLineEdit_99["font"] = ft
App.GLineEdit_99["fg"] = "#333333"
App.GLineEdit_99["justify"] = "center"
App.GLineEdit_99["text"] = "Entry"
App.GLineEdit_99.place(x=270, y=110, width=271, height=36)

GButton_972 = tk.Button(root)
GButton_972["bg"] = "#111112"
ft = tkFont.Font(family='Times', size=10)
GButton_972["font"] = ft
GButton_972["fg"] = "#cc0000"
GButton_972["justify"] = "center"
GButton_972["text"] = "Delete"
GButton_972.place(x=250, y=220, width=94, height=34)
GButton_972["command"] = delete

def delete():
    conn = mysql.connector.connect(host="localhost", user="root",
password="Rr2163?!", database="car_rentals")
    cur = conn.cursor()

    Car_no = App.GLineEdit_99.get()

    cur.execute("delete from car_details where car_no =
'{}'".format(Car_no))

    try:
        conn.commit()
        conn.close()
        messagebox.showinfo("Success", "Car Detail Removed Successfully!")

    except:
        messagebox.showinfo("Error", "Couldn't delete car detail!")

```

Add.py

```

import tkinter as tk
import tkinter.font as tkFont
import mysql.connector
from tkinter import messagebox

class App:

    root = None
    GLineEdit_396 = None
    GLineEdit_995 = None
    GLineEdit_764 = None
    GLineEdit_466 = None
    GLineEdit_427 = None
    GLineEdit_580 = None

    def __init__(self, root):
        #setting title
        root.title("Add")
        #setting window size
        width=600
        height=500

```

```

        screenwidth = root.winfo_screenwidth()
        screenheight = root.winfo_screenheight()
        alignstr = '%dx%d+%d+%d' % (width, height, (screenwidth - width) /
2, (screenheight - height) / 2)
        root.geometry(alignstr)
        root.resizable(width=False, height=False)

        GLabel_923=tk.Label(root)
        GLabel_923["bg"] = "#170b0b"
        ft = tkFont.Font(family='Times',size=10)
        GLabel_923["font"] = ft
        GLabel_923["fg"] = "#e91818"
        GLabel_923["justify"] = "center"
        GLabel_923["text"] = "Car Number"
        GLabel_923.place(x=50,y=100,width=70,height=25)

        GLabel_763=tk.Label(root)
        GLabel_763["bg"] = "#230707"
        ft = tkFont.Font(family='Times',size=10)
        GLabel_763["font"] = ft
        GLabel_763["fg"] = "#e21616"
        GLabel_763["justify"] = "center"
        GLabel_763["text"] = "Car Type"
        GLabel_763.place(x=50,y=160,width=70,height=25)

        GLabel_388=tk.Label(root)
        GLabel_388["bg"] = "#240707"
        ft = tkFont.Font(family='Times',size=10)
        GLabel_388["font"] = ft
        GLabel_388["fg"] = "#e31818"
        GLabel_388["justify"] = "center"
        GLabel_388["text"] = "Your ID"
        GLabel_388.place(x=50,y=220,width=70,height=25)

        GLabel_945=tk.Label(root)
        GLabel_945["bg"] = "#100b0b"
        ft = tkFont.Font(family='Times',size=10)
        GLabel_945["font"] = ft
        GLabel_945["fg"] = "#d42020"
        GLabel_945["justify"] = "center"
        GLabel_945["text"] = "Rental Price"
        GLabel_945.place(x=50,y=280,width=70,height=25)

        GLabel_772=tk.Label(root)
        GLabel_772["bg"] = "#060303"
        ft = tkFont.Font(family='Times',size=10)
        GLabel_772["font"] = ft
        GLabel_772["fg"] = "#ea2222"
        GLabel_772["justify"] = "center"
        GLabel_772["text"] = "Tariff"
        GLabel_772.place(x=50,y=340,width=70,height=25)

        GLabel_727=tk.Label(root)
        GLabel_727["bg"] = "#230606"
        ft = tkFont.Font(family='Times',size=10)
        GLabel_727["font"] = ft
        GLabel_727["fg"] = "#d62323"
        GLabel_727["justify"] = "center"
        GLabel_727["text"] = "Car Name"
        GLabel_727.place(x=50,y=400,width=70,height=25)

```

```
App.GLineEdit_580=tk.Entry(root)
App.GLineEdit_580["borderwidth"] = "1px"
ft = tkFont.Font(family='Times',size=10)
App.GLineEdit_580["font"] = ft
App.GLineEdit_580["fg"] = "#333333"
App.GLineEdit_580["justify"] = "center"
App.GLineEdit_580["text"] = ""
App.GLineEdit_580.place(x=160,y=400,width=200,height=25)

App.GLineEdit_396=tk.Entry(root)
App.GLineEdit_396["borderwidth"] = "1px"
ft = tkFont.Font(family='Times',size=10)
App.GLineEdit_396["font"] = ft
App.GLineEdit_396["fg"] = "#333333"
App.GLineEdit_396["justify"] = "center"
App.GLineEdit_396["text"] = ""
App.GLineEdit_396.place(x=160,y=100,width=200,height=25)

App.GLineEdit_995=tk.Entry(root)
App.GLineEdit_995["borderwidth"] = "1px"
ft = tkFont.Font(family='Times',size=10)
App.GLineEdit_995["font"] = ft
App.GLineEdit_995["fg"] = "#333333"
App.GLineEdit_995["justify"] = "center"
App.GLineEdit_995["text"] = ""
App.GLineEdit_995.place(x=160,y=160,width=200,height=25)

App.GLineEdit_764=tk.Entry(root)
App.GLineEdit_764["borderwidth"] = "1px"
ft = tkFont.Font(family='Times',size=10)
App.GLineEdit_764["font"] = ft
App.GLineEdit_764["fg"] = "#333333"
App.GLineEdit_764["justify"] = "center"
App.GLineEdit_764["text"] = ""
App.GLineEdit_764.place(x=160,y=220,width=200,height=25)

App.GLineEdit_466=tk.Entry(root)
App.GLineEdit_466["borderwidth"] = "1px"
ft = tkFont.Font(family='Times',size=10)
App.GLineEdit_466["font"] = ft
App.GLineEdit_466["fg"] = "#333333"
App.GLineEdit_466["justify"] = "center"
App.GLineEdit_466["text"] = ""
App.GLineEdit_466.place(x=160,y=280,width=200,height=25)

App.GLineEdit_427=tk.Entry(root)
App.GLineEdit_427["borderwidth"] = "1px"
ft = tkFont.Font(family='Times',size=10)
App.GLineEdit_427["font"] = ft
App.GLineEdit_427["fg"] = "#333333"
App.GLineEdit_427["justify"] = "center"
App.GLineEdit_427["text"] = ""
App.GLineEdit_427.place(x=160,y=340,width=200,height=25)

GButton_921=tk.Button(root)
GButton_921["bg"] = "#1d1919"
ft = tkFont.Font(family='Times',size=10)
GButton_921["font"] = ft
GButton_921["fg"] = "#e21f1f"
GButton_921["justify"] = "center"
GButton_921["text"] = "Add"
```



```

        GButton_921.place(x=440,y=190,width=96,height=91)
        GButton_921["command"] = insert_details

def insert_details():

    conn =
mysql.connector.connect(host="localhost",user="root",password="Rr2163?!",da
tabase="car_rentals")
    cur = conn.cursor()

    car_no = App.GLineEdit_396.get()
    car_type = App.GLineEdit_995.get()
    agent_id = App.GLineEdit_764.get()
    rental_price = App.GLineEdit_466.get()
    tariff = App.GLineEdit_427.get()
    car_name = App.GLineEdit_580.get()

    cur.execute("insert into car_details
values('{}','{}','{}','{}','{}','{}')".format(car_no,car_type,agent_id,rent
al_price,tariff,car_name))
    try:
        conn.commit()
        conn.close()
        messagebox.showinfo("Success","Car Details Inserted Successfully!")
    except:
        messagebox.showinfo("Error","Couldn't add Car to our database")

```

View.py

```

import tkinter as tk
import tkinter.font as tkFont
import mysql.connector
from tkinter import *

class App:

    GListBox_400 = None

    def __init__(self, root):
        #setting title
        root.title("undefined")
        #setting window size
        width=600
        height=500
        screenwidth = root.winfo_screenwidth()
        screenheight = root.winfo_screenheight()
        alignstr = '%dx%d+%d+%d' % (width, height, (screenwidth - width) /
2, (screenheight - height) / 2)
        root.geometry(alignstr)
        root.resizable(width=False, height=False)

        GButton_949=tk.Button(root)
        GButton_949["bg"] = "#210d0d"
        ft = tkFont.Font(family='Times',size=10)
        GButton_949["font"] = ft
        GButton_949["fg"] = "#d71111"
        GButton_949["justify"] = "center"
        GButton_949["text"] = "View"
        GButton_949.place(x=250,y=30,width=70,height=25)

```

```

GButton_949["command"] = View

App.GListBox_400=tk.Listbox(root)
App.GListBox_400["borderwidth"] = "1px"
ft = tkFont.Font(family='Times',size=10)
App.GListBox_400["font"] = ft
App.GListBox_400["fg"] = "#333333"
App.GListBox_400["justify"] = "center"
App.GListBox_400.place(x=20,y=80,width=558,height=403)

def View():

    conn = mysql.connector.connect(host="localhost", user="root",
password="Rr2163?!", database="car_rentals")
    cur = conn.cursor()

    cur.execute("select * from car_details")
    c = cur.fetchall()

    App.GListBox_400.insert(END," Car No          Car Type          Agent ID
RentalPrice      Tariff      Car Name")
    App.GListBox_400.insert(END,"-----
-----")

    for i in c:
        App.GListBox_400.insert(END,str(i[0]) + "      " + str([i[1]]) + "
" + str(i[2]) + "      " + str(i[3]) + "      " + str(i[4]) + "      " +
str(i[5]))

    conn.commit()
    conn.close()

```

Sample Outputs:

Main Page:



New Agent Registration:

undefined

Name

Ram

Email ID

ram@432

Phone

456881


Desired Password

scott

Register

Exit

Agent Login:

 EXISTING AGENT LOGIN

Email ID

rer@343

Password


Submit

Exit

Agent Add / Update / Delete/View:



ADD:

 Add

— □ ×

Car Number	13281
Car Type	Sedan
Your ID	343
Rental Price	50000
Tariff	9000
Car Name	Mitsuha

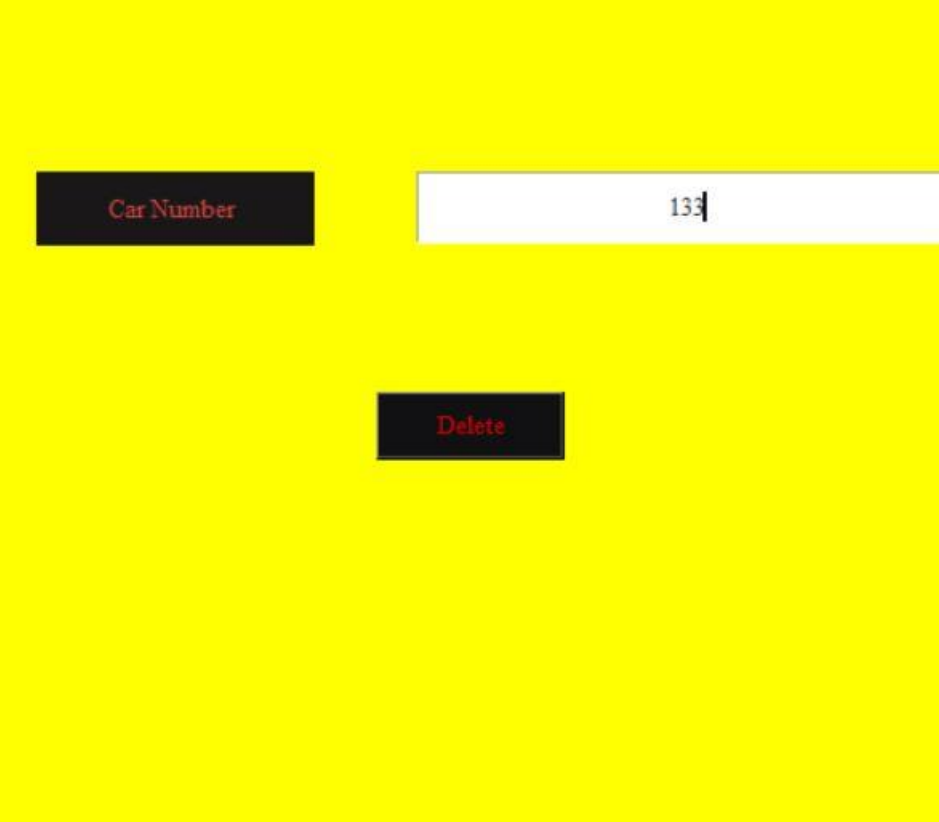
Add

Update:

undefined

Car Number	Car Type	Rental Price	Tariff	Car Name
Enter Car Number	<input type="text" value="133"/>			
Enter New Value	<input type="text" value="4000"/>			

Delete:



Delete

Car Number

133

Delete

Agent View:

The image shows a web browser window titled "Agent View". It features a yellow header bar with a "View" button in the center. Below the header is a table with six columns: Car No, Car Type, Agent ID, RentalPrice, Tariff, and Car Name. The table contains five rows of data. The window has standard browser controls (back, forward, refresh, address bar) and a yellow border.

Car No	Car Type	Agent ID	RentalPrice	Tariff	Car Name
133	[Sedan]	0_sne@132	30000	20000	Honda
156	[SUV]	132	40000	20000	Hyundai
15678	[RUX]	0_sne@132	40900	15001	Maruti
13456	[Sedan]	0_sne@132	40000	60000	Honda
13281	[Sedan]	343	50000	9000	Mitsuha

Customer View:

 CUSTOMER VIEW — □ ×

Name

Resds

Submit

Phone Number

987642

Exit

Maximum Expected Price

70000

Car No	Car Type	RentalPrice	Tariff	Car Name	AgentName	AgentPhoneNum	AgentEmail
--------	----------	-------------	--------	----------	-----------	---------------	------------

Result:

The project Car Rental System using pycharm and mysql has been designed for inserting, updating , deleting, viewing the required details has been created.

Appendix:

Report on Pycharm:

PyCharm is an integrated development environment (IDE) used in computer programming, specifically for the Python language. It is developed by the Czech company JetBrains (formerly known as IntelliJ). It provides code analysis, a graphical debugger, an integrated unit tester, integration with version control systems (VCSes), and supports web development with Django as well as data science with Anaconda.

PyCharm is cross-platform, with Windows, macOS and Linux versions. The Community Edition is released under the Apache License, and there is also Professional Edition with extra features – released under a proprietary license.

Features of Pycharm:

- Coding assistance and analysis, with code completion, syntax and error highlighting, linter integration, and quick fixes
- Project and code navigation: specialized project views, file structure views and quick jumping between files, classes, methods and usages
- Python refactoring: includes rename, extract method, introduce variable, introduce constant, pull up, push down and others
- Support for web frameworks: Django, web2py and Flask [professional edition only]
- Integrated Python debugger
- Integrated unit testing, with line-by-line code coverage
- Google App Engine Python development [professional edition only]
- Version control integration: unified user interface for Mercurial, Git, Subversion, Perforce and CVS with change lists and merge
- Support for scientific tools like matplotlib, numpy and scipy [professional edition only]

It competes mainly with a number of other Python-oriented IDEs, including Eclipse's PyDev, and the more broadly focused Komodo IDE.

Report on Tkinter GUI:

Out of all GUI methods, Tkinter is the most commonly used method. It is a standard python interface to the Tk GUI toolkit shipped with python. Python with tkinter is the fastest and easiest way to create the GUI Applications.

Tkinter supports a range of Tcl/Tk versions, built either with or without thread support. The official Python binary release bundles Tcl/Tk 8.6 threaded. See the source code for the `_tkinter` module for more information about supported versions.

Tkinter is not a thin wrapper, but adds a fair amount of its own logic to make the experience more pythonic. This documentation will concentrate on these additions and changes, and refer to the official Tcl/Tk documentation for details that are unchanged.

Architecture

Tcl

Tcl is a dynamic interpreted programming language, just like Python. Though it can be used on its own as a general-purpose programming language, it is most commonly embedded into C applications as a scripting engine or an interface to the Tk toolkit

Tk

Tk is a Tcl package implemented in C that adds custom commands to create and manipulate GUI widgets. Each Tk object embeds its own Tcl interpreter instance with Tk loaded into it. Tk's widgets are very customizable, though at the cost of a dated appearance. Tk uses Tcl's event queue to generate and process GUI events.

Ttk

Themed Tk (Ttk) is a newer family of Tk widgets that provide a much better appearance on different platforms than many of the classic Tk widgets. Ttk is distributed as part of Tk, starting with Tk version 8.5. Python bindings are provided in a separate module, `tkinter.ttk`.

Tkinter Modules

```
from tkinter import *
```

```
from tkinter import ttk
```

```
class tkinter.Tk(screenName=None, baseName=None, className='Tk', useTk=1)
```

The Tk class is instantiated without arguments. This creates a toplevel widget of Tk which usually is the main window of an application. Each instance has its own associated Tcl interpreter.

```
tkinter.Tcl(screenName=None, baseName=None, className='Tk', useTk=0)
```

The `Tcl()` function is a factory function which creates an object much like that created by the `Tk` class, except that it does not initialize the `Tk` subsystem. This is most often useful when driving the `Tcl` interpreter in an environment where one doesn't want to create extraneous toplevel windows, or where one cannot (such as Unix/Linux systems without an X server). An object created by the `Tcl()` object can have a Toplevel window created (and the `Tk` subsystem initialized) by calling its `loadtk()` method.

The modules that provide `Tk` support include:

`tkinter`

Main `Tkinter` module.

`tkinter.colorchooser`

Dialog to let the user choose a color.

`tkinter.commondialog`

Base class for the dialogs defined in the other modules listed here.

`tkinter.filedialog`

Common dialogs to allow the user to specify a file to open or save.

`tkinter.font`

Utilities to help work with fonts.

`tkinter.messagebox`

Access to standard `Tk` dialog boxes.

`tkinter.scrolledtext`

Text widget with a vertical scroll bar built in.

`tkinter.simpledialog`

Basic dialogs and convenience functions.

`tkinter.ttk`

Themed widget set introduced in `Tk` 8.5, providing modern alternatives for many of the classic widgets in the main `tkinter` module.

Additional modules:

`_tkinter`

A binary module that contains the low-level interface to `Tcl/Tk`. It is automatically imported by the main `tkinter` module, and should never be used directly by application programmers. It is usually a shared library (or `DLL`), but might in some cases be statically linked with the `Python` interpreter.

`idlelib`

Python's Integrated Development and Learning Environment (IDLE). Based on tkinter.

tkinter.constants

Symbolic constants that can be used in place of strings when passing various parameters to Tkinter calls. Automatically imported by the main tkinter module.

tkinter.dnd

(experimental) Drag-and-drop support for tkinter. This will become deprecated when it is replaced with the Tk DND.

tkinter.tix

(deprecated) An older third-party Tcl/Tk package that adds several new widgets. Better alternatives for most can be found in tkinter.ttk.

turtle

Turtle graphics in a Tk window.

References:

<https://docs.python.org/3/library/tkinter.html>

<https://realpython.com/python-gui-tkinter/>

<https://www.jetbrains.com/help/pycharm/quick-start-guide.html>