



PES UNIVERSITY

(Established under Karnataka Act No. 16 of 2013)

100 Feet Ring Road, BSK III Stage, Bengaluru-560 085

Department of Computer Science & Engineering

Title: Car Rental Management System

**Team member details: PES2UG21CS158-Deepika Indran, PES2UG21CS251-Lakshmi
Narayanan**

Abstract

Bon Voyage Rentals is an app which allows you to rent or return a car. The GUI of the app is designed with the help of the tkinter module in python and the list of available cars and their details are stored in MySQL.

On opening the app, you can choose to rent or return a car. Once you choose to rent, you can select your desired filters and accordingly view the available cars. On confirming your selection, the database automatically lists that particular car as ‘unavailable’ for future customers. If you choose to return a previously rented car, the app adds this particular car as a new record in the database.

Table of Contents

1. Introduction
2. Design/Implementation
3. Testing
4. Result and Analysis
5. Conclusions & future enhancements
6. References

Introduction

Bon Voyage Rentals is a car rental service app which helps ease the process of car rentals and returns, in Bengaluru. With the advancement of new technologies, the city is witnessing a shifting culture that calls for an equal focus on work and leisure. With youngsters increasingly viewing driving as a pleasure rather than a chore, car rentals have emerged as the perfect investment-free solution. Bon Voyage Rentals is a hassle-free and user-friendly app which provides an effective system through which you can choose your desired car within minutes.

The cars listed on our site come with all India permits, Road Side Assistance (RSA), and include vehicle insurance. It is extremely easy to pick the rental cars as there are multiple pickup points across the city. While small cars allow you to manoeuvre through city traffic and are ideal for a daily commute, sedans and SUVs offer more comfort. We make it possible for you to pick your car based on your travelling needs.

Design/Implementation

```
from tkinter import*
from tkinter import ttk
import mysql.connector
connection=mysql.connector.connect(host='localhost',user='root',password='munich24',database='computerproject')
cursor=connection.cursor()
def screen1():
    global root
    root=Tk()
    root.title("Car Rental Management System")
    root.geometry("1200x800")
    bg=PhotoImage(file=r"C:\Users\91855\Desktop\Carrental\image4.png")
    my_label=Label(root,image=bg)
    my_label.pack()
    headingFrame1 = Frame(root,bg="#FFBB00",bd=5)
    headingFrame1.place(relx=0.2,rely=0.1,relwidth=0.6,relheight=0.16)
    headingLabel = Label(headingFrame1, text="Welcome to Bon Voyage Rentals!", bg="black", fg='white', font=('Courier',25))
    headingLabel.place(relx=0,rely=0, relwidth=1, relheight=1)
```

```
btn1=Button(root,text="Rent           a
car",command=filters,bg="black",fg='white',font=('arial',15,'bold')).place(relx=0.28,rely=0.4,
relwidth=0.45,relheight=0.1)

btn2=Button(root,text="Return         a
car",command=returns,bg="black",fg='white',font=('arial',15,'bold')).place(relx=0.28,rely=0.6,
relwidth=0.45,relheight=0.1)

root.mainloop()
```

```
def filters():

    global screen4,tree,e1

    root.destroy()

    screen4= Tk()

    screen4.title("Filters")

    screen4.geometry("1200x600")

    bg2=PhotoImage(file=r"C:\Users\91855\Desktop\Carrental\image8.png")

    my_label2=Label(screen4,image=bg2).pack()

        lb1=Label(screen4,text="Select the filter you would like to apply",bg="black",
fg="white",font=("Segoe Print",20,'bold')).place(relx=0.28, rely=0.01)

    global var1,var2,var3,var4,var18,cb1,cb2,cb3,cb4

                                var1,var2,var3,var4,var18
=IntVar(screen4),IntVar(screen4),IntVar(screen4),IntVar(screen4),StringVar(screen4)
```

```
cb1= Checkbutton(screen4,text="Pick-up Location",command=showfilter,  
variable=var1,font=('Segoe Print',14,'bold'))
```

```
cb1.place(relx=0.12, rely=0.22)
```

```
cb2= Checkbutton(screen4,text="With/Without Driver",  
command=showfilter,variable=var2,font=('Segoe Print',14,'bold'))
```

```
cb2.place(relx=0.12, rely=0.32)
```

```
cb3= Checkbutton(screen4,text="Price Range",  
command=showfilter,variable=var3,font=('Segoe Print',14,'bold'))
```

```
cb3.place(relx=0.12, rely=0.42)
```

```
cb4= Checkbutton(screen4,text="Seater Capacity",  
command=showfilter,variable=var4,font=('Segoe Print',14,'bold'))
```

```
cb4.place(relx=0.12, rely=0.52)
```

```
#Assign the width, minwidth, and anchor to the respective columns
```

```
tree = ttk.Treeview(screen4, columns=("Car", "Seater Capacity", "Price per day", "Pick-up Location", "Driver Needed?", "Availability"), show="headings")
```

```
tree["columns"] =("Car", "Seater Capacity", "Price per day", "Pick-up Location", "Driver Needed?", "Availability")
```

```
tree.column("Car",width=150,minwidth=30,anchor="s")
```

```
tree.column("Seater Capacity", width=100, minwidth=50, anchor="s")
tree.column("Price per day", width=100, minwidth=50, anchor="s")
tree.column("Pick-up Location", width=100, minwidth=50, anchor="s")
tree.column("Driver Needed?", width=100, minwidth=50, anchor="s")
tree.column("Availability", width=100, minwidth=50, anchor="s")
tree.place(relx=0.35, rely=0.2)
```

#Assign the heading names

```
tree.heading("Car",text="Car",anchor="s")
tree.heading("Seater Capacity", text="Seater Capacity", anchor="s")
tree.heading("Price per day", text="Price per day", anchor="s")
tree.heading("Pick-up Location", text="Pick-up Location", anchor="s")
tree.heading("Driver Needed?", text="Driver Needed?", anchor="s")
tree.heading("Availability", text="Availability", anchor="s")
```

```
e1=Entry(screen4,width=30,textvariable=var18).place(relx=0.70,rely=0.86)
```

```
lb18=Label(screen4,text="Please enter the name of the car you'd like to
select:",font=('Arial',14,'bold')).place(relx=0.29,rely=0.85)
```

```
btn6=      Button(screen4,      text=      "SELECT",command=selection,
font=('Calibri',15)).place(relx=0.88,rely=0.85)
```

```
        btn11= Button(screen4, text= "Try another filter",command=deselect,  
font=('Calibri',15)).place(relx=0.81,rely=0.65)
```

```
screen4.mainloop()
```

```
def showfilter():
```

```
    if var1.get()==1:
```

```
        pickup()
```

```
    if var2.get()==1:
```

```
        driver()
```

```
    if var3.get()==1:
```

```
        price()
```

```
    if var4.get()==1:
```

```
        seater()
```

```
def deselect():
```

```
    for item in tree.get_children():
```

```
        tree.delete(item)
```

```
        cb1.deselect()
```

```
        cb2.deselect()
```

```
        cb3.deselect()
```

```
cb4.deselect()

def pickup():

    global screen5,screen4

    screen5=Toplevel(screen4)

    global var5,var6,var7,var8,var9,var10

    var5,var6,var7,var8,var9,var10=IntVar(screen5),IntVar(screen5),IntVar(screen5),IntVar(screen5),
    IntVar(screen5),IntVar(screen5),

    screen5.title("Pick-up Location")

    screen5.geometry("800x500")

    bg3=PhotoImage(file=r"C:\Users\91855\Desktop\Carrental\image9.png")

    my_label3=Label(screen5,image=bg3).pack()

    lb2=Label(screen5,text="Please select the pick-up location",bg="black",
    fg="white",font=("Segoe Print",20,'bold')).place(relx=0.2, rely=0.01)

    cb5=           Checkbutton(screen5,text="Bellandur",
    variable=var5,font=('Calibri',14,'bold')).place(relx=0.15, rely=0.32)

    cb6=           Checkbutton(screen5,text="Shivajinagar",
    variable=var6,font=('Calibri',14,'bold')).place(relx=0.4, rely=0.32)

    cb7=           Checkbutton(screen5,text="Koramangala",
    variable=var7,font=('Calibri',14,'bold')).place(relx=0.67, rely=0.32)
```

```
cb8= Checkbutton(screen5,text="Bananashankari",
variable=var8,font=('Calibri',14,'bold')).place(relx=0.15, rely=0.52)
```

```
cb9= Checkbutton(screen5,text="Indiranagar",
variable=var9,font=('Calibri',14,'bold')).place(relx=0.4, rely=0.52)
```

```
cb10= Checkbutton(screen5,text="Hsr Layout",
variable=var10,font=('Calibri',14,'bold')).place(relx=0.67, rely=0.52)
```

```
btn3= Button(screen5, text= "OK",command=show,
font=('Calibri',15)).place(relx=0.9,rely=0.8)
```

```
screen5.mainloop()
```

```
def driver():
```

```
global screen6,screen4
```

```
screen6= Toplevel(screen4)
```

```
global var11,var12
```

```
var11,var12=IntVar(screen6),IntVar(screen6)
```

```
screen6.title("With/Without Driver")
```

```
screen6.geometry("600x400")
```

```
bg4=PhotoImage(file=r"C:\Users\91855\Desktop\Carrental\image9.png")
```

```
my_label4=Label(screen6,image=bg4).pack()
```

```
lb3=Label(screen6,text="Would you want a driver?",bg="black", fg="white",font=("Segoe Print",20,'bold')).place(relx=0.2, rely=0.01)

cb11=           Checkbutton(screen6,text="Yes",
variable=var11,font=('Calibri',20,'bold')).place(relx=0.18, rely=0.3)

cb12=           Checkbutton(screen6,text="No",
variable=var12,font=('Calibri',20,'bold')).place(relx=0.68, rely=0.3)

btn4=           Button(screen6,      text=      "OK",command=show,
font=('Calibri',15)).place(relx=0.9,rely=0.8)

screen6.mainloop()
```

```
def price():

    global screen7

    screen7=Toplevel(screen4)

    global var13,var14,var15

    var13,var14,var15=IntVar(screen7),IntVar(screen7),IntVar(screen7)

    screen7.title("Price Range")

    screen7.geometry("600x400")

    bg5=PhotoImage(file=r"C:\Users\91855\Desktop\Carrental\image9.png")

    my_label5=Label(screen7,image=bg5).pack()

    lb4=Label(screen7,text="Choose your price range",bg="black", fg="white",font=("Segoe Print",20,'bold')).place(relx=0.22, rely=0.01)
```

```
cb13= Checkbutton(screen7,text="2000-3000",
variable=var13,font=('Calibri',14,'bold')).place(relx=0.40, rely=0.3)

cb14= Checkbutton(screen7,text="3000-4000",
variable=var14,font=('Calibri',14,'bold')).place(relx=0.40, rely=0.5)

cb15= Checkbutton(screen7,text="4000-5000",
variable=var15,font=('Calibri',14,'bold')).place(relx=0.40, rely=0.7)

btn5= Button(screen7, text= "OK",command=show,
font=('Calibri',15)).place(relx=0.9,rely=0.8)

screen7.mainloop()
```

```
def seater():

    global screen8

    screen8=Toplevel(screen4)

    global var16,var17,var18

    var16,var17,var18 =IntVar(screen8),IntVar(screen8),IntVar(screen8)

    screen8.title("Seater Capacity")

    screen8.geometry("600x400")

    bg6=PhotoImage(file=r"C:\Users\91855\Desktop\Carrental\image9.png")

    my_label6=Label(screen8,image=bg6).pack()
```

```
lb5=Label(screen8,text="Seater Capacity",bg="black", fg="white",font=("Segoe Print",20,"bold")).place(relx=0.3, rely=0.01)

cb16= Checkbutton(screen8,text="5-seater",
variable=var16,font=('Calibri',14,'bold')).place(relx=0.40, rely=0.3)

cb17= Checkbutton(screen8,text="7-seater",
variable=var17,font=('Calibri',14,'bold')).place(relx=0.40, rely=0.5)

cb18= Checkbutton(screen8,text="8-seater",
variable=var18,font=('Calibri',14,'bold')).place(relx=0.40, rely=0.7)

btn5= Button(screen8, text= "OK",command=show,
font=('Calibri',15)).place(relx=0.9,rely=0.8)

screen8.mainloop()
```

```
#insert values into the window from MySQL table
```

```
def show():

    global m,sqlquery

    if var1.get()==1:

        screen5.destroy()

        if var5.get()==1:

            m="""Bellandur"""

        elif var6.get()==1:
```

```
m=""Shivajinagar"

elif var7.get()==1:

    m=""Koramangala"

elif var8.get()==1:

    m=""Bananashankari"

elif var9.get()==1:

    m=""Indiranagar"

elif var10.get()==1:

    m=""HSR Layout"

sqlquery="select* from cars where pickup=" + m

elif var2.get()==1:

    screen6.destroy()

    if var11.get()==1:

        m=""Yes"

    else:

        m=""No"

    sqlquery="select* from cars where driver=" + m
```

```
elif var3.get()==1:  
  
    screen7.destroy()  
  
if var13.get()==1:  
  
    p,q="2000","3000"  
  
elif var14.get()==1:  
  
    p,q="3000","4000"  
  
elif var15.get()==1:  
  
    p,q="4000","5000"  
  
sqlquery="select* from cars where price between "+p+" and "+q  
  
  
  
elif var4.get()==1:  
  
    screen8.destroy()  
  
if var16.get()==1:  
  
    m="5"  
  
elif var17.get()==1:  
  
    m="7"  
  
elif var18.get()==1:  
  
    m="8"  
  
sqlquery="select* from cars where seater=" + m
```

```
cursor.execute(sqlquery)

rows=cursor.fetchall()

for x in rows:

    tree.insert("", 'end', values=x)

def selection():

    r="""+var18.get()"""

    remdata="update cars set Availability= 'Unavailable' where carname='"+r

    cursor.execute(remdata)

    connection.commit()

    selectiontext="You have successfully rented " + r +" !"

lb19=Label(screen4,text=selectiontext,font=('Arial',14,'bold'),bg="black",fg="white").place(relx
=0.5,rely=0.75)

def returns():

    root.destroy()

screen3=Tk()

screen3.geometry("1200x800")
```

```
screen3.title("Car Returns")

bg1=PhotoImage(file=r"C:\Users\91855\Desktop\Carrental\image10.png")

my_label1=Label(screen3,image=bg1)

my_label1.pack()

headingFrame2 = Frame(screen3,bg="black",bd=5)

headingFrame2.place(relx=0.42,rely=0.2,relwidth=0.18,relheight=0.09)

frame3=Frame(screen3,bg="black")

frame3.place(relx=0.23,rely=0.33,relwidth=0.55,relheight=0.25)

lbb=Label(headingFrame2,text="Return a car",bg="white",fg="black",font=("Segoe
Print",20,'bold')).place(relx=0.05, rely=0.05)

def showreturn():

    global z

    z=cars.get()

    global y

    y=area.get()

    mylabel=Label(screen3,text= "You returned a "+cars.get()+" at "+area.get()
,font=('Calibri',20,'bold'),fg="white",bg="black").place(relx=0.32,rely=0.70)
```

```
cursor.execute("INSERT INTO cars(carname,pickup) values(%s,%s)", (z,y))
```

```
connection.commit()
```

```
cars=StringVar()
```

```
area=StringVar()
```

```
drop=OptionMenu(frame3,cars,"WagonR","Honda City","Swift Dzire","Innova","Xylo","Creta","i20","Jeep Compass","Corolla Altis","Verna","Astar","Duster","i10","Fortuner").place(relx=0.21,rely=0.27,relwidth=0.05,relheight=0.05)
```

```
lb15=Label(frame3,text="Choose your car",font=("Segoe Print",18,'bold'),fg="black",bg="white").place(relx=0.27,rely=0.13,relheight=0.3,relwidth=0.5)
```

```
drop1=OptionMenu(frame3,area,"Bellandur","Shivajinagar","Koramangala","Indiranagar","HSR Layout","Bananashankari").place(relx=0.02,rely=0.69,relwidth=0.05,relheight=0.05)
```

```
lb16=Label(frame3,text="Choose the area you wish to return at",font=("Segoe Print",18,'bold'),fg="black",bg="white").place(relx=0.08,rely=0.52,relwidth=0.85,relheight=0.4)
```

```
mybutton=Button(screen3,text="Show Selection",command=showreturn).place(relx=0.7,rely=0.6)
```

```
e=Entry(screen3,width=30).place(relx=0.30,rely=0.91)

lb17=Label(screen3,text="Help us
improve!",font=('Arial',14,'bold')).place(relx=0.15,rely=0.9)

screen3.mainloop()

screen1()
```

Testing

```
C:\Users\91855\Desktop\Carrental>py -m pytest test_one.py
```

On testing the code using pytest, the following result was obtained:

```
platform win32 -- Python 3.9.5, pytest-6.2.5, py-1.11.0, pluggy-1.0.0
rootdir: C:\Users\91855\Desktop\Carrental
collected 1 item

test_one.py . [■

-----
```

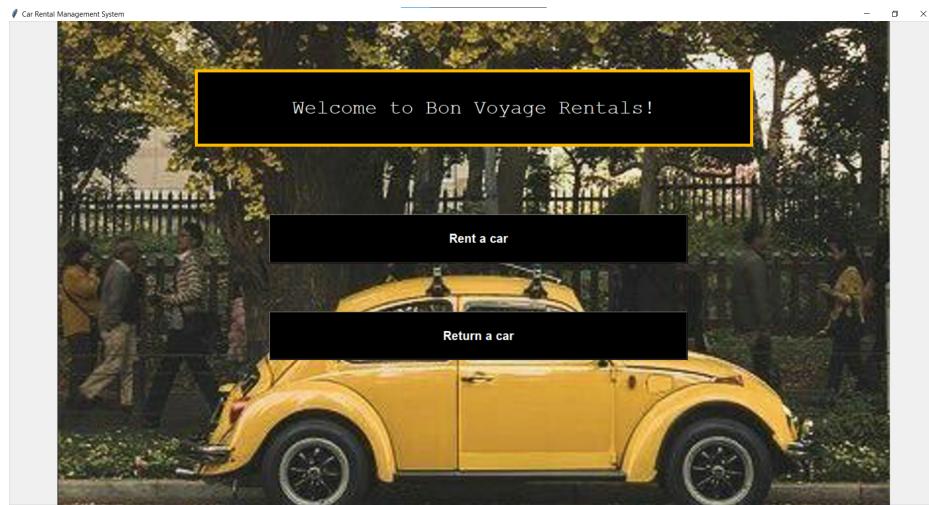


```
1 passed, 1 warning in 25.94s
```

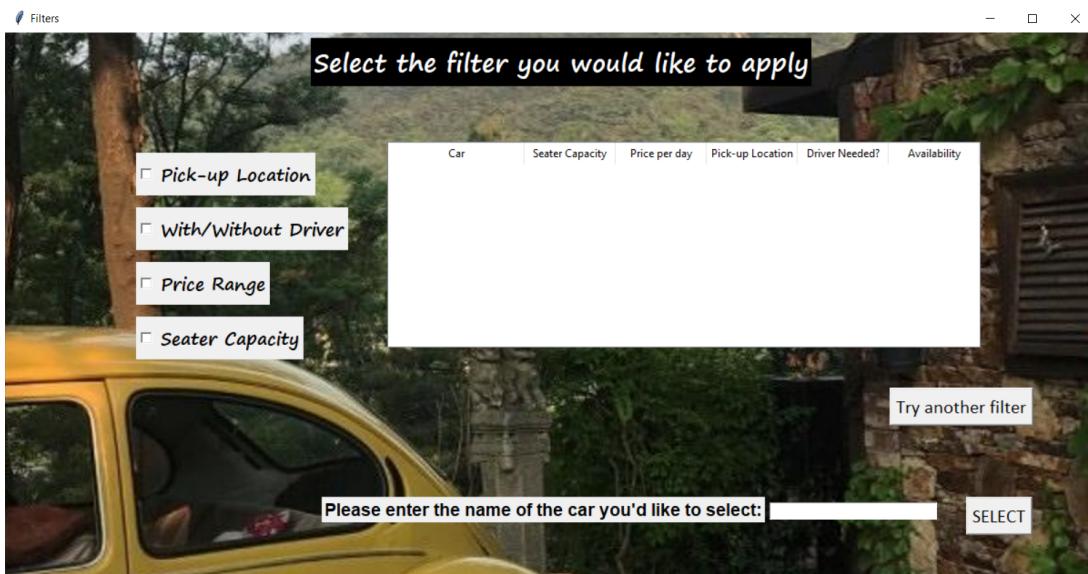
The code runs successfully and no error was thrown.

Result and Analysis:

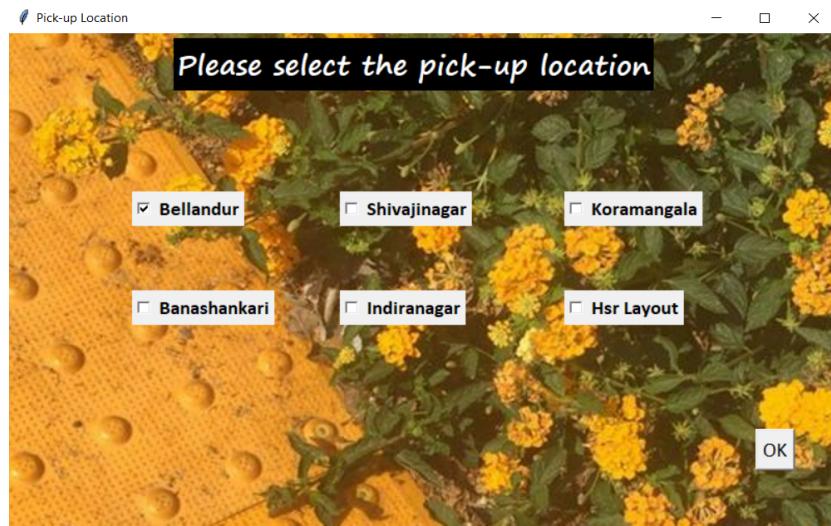
- 1) The welcome page displays the name of the app and options to rent or return a car:



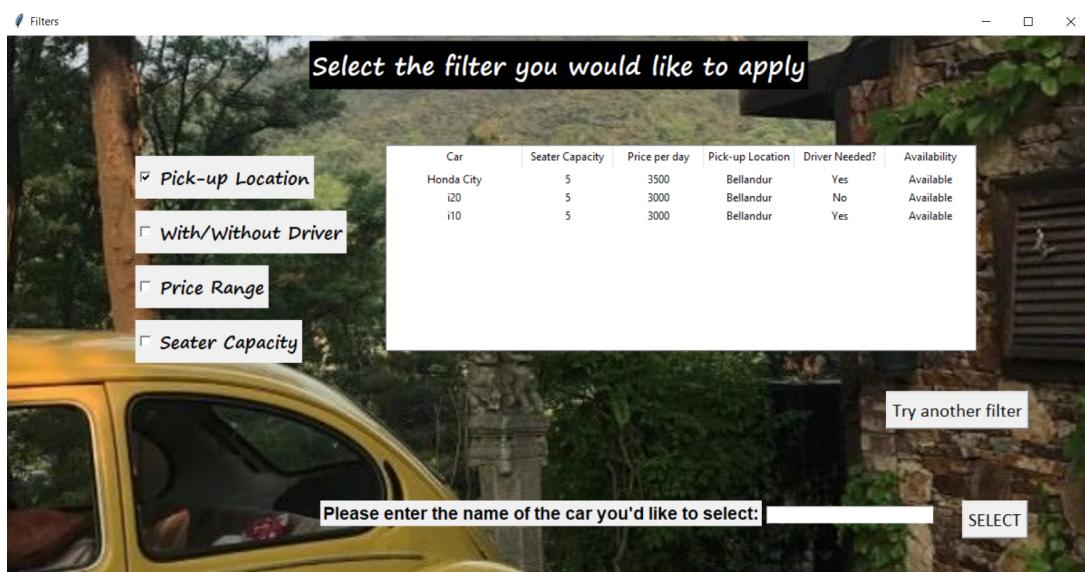
- 2) On clicking the 'Rent' button, a new window opens with a list of checkboxes. This allows the user to select which filter they would like to apply:



- 3) For example, if the ‘Pick-up Location’ filter is chosen, a list of available pick-up points is displayed as checkbuttons, for the user to choose from:



- 4) Once the user selects their preferred pick-up location (eg: ‘Bellandur’ is selected), a list of available cars and their details is displayed in the form of a table:



The image shows a "Filters" interface. On the left, there is a sidebar with three checkboxes: "Pick-up Location" (checked), "With/Without Driver", and "Price Range". Below these is another checkbox for "Seater Capacity". To the right of the sidebar is a table displaying car details. At the bottom of the interface, there is a search bar with the placeholder text "Please enter the name of the car you'd like to select:" and a "SELECT" button.

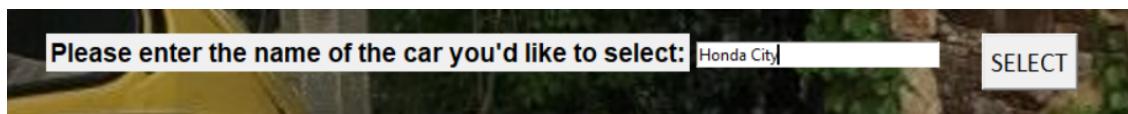
Car	Seater Capacity	Price per day	Pick-up Location	Driver Needed?	Availability
Honda City	5	3500	Bellandur	Yes	Available
i20	5	3000	Bellandur	No	Available
i10	5	3000	Bellandur	Yes	Available

- 5) This list is selected and displayed from our database in MySQL. It can be seen that all the cars available at the pick-up location ‘Bellandur’, were displayed in the previous screenshot:

carname	seater	price	pickup	driver	Availability
WagonR	5	2000	Shivajinagar	Yes	Available
Honda City	5	3500	Bellandur	Yes	Available
Swift Dzire	5	3500	Koramangala	No	Available
Innova	7	4500	Koramangala	Yes	Available
Baleno	5	4500	Banashankari	Yes	Available
Xylo	7	4500	Indiranagar	Yes	Available
Creta	5	4000	HSR Layout	No	Available
i20	5	3000	Bellandur	No	Available
Jeep Compass	8	5000	Indiranagar	No	Available
Corolla Altis	5	3200	Indiranagar	No	Available
Verna	5	3700	Shivajinagar	No	Available
Astar	5	2500	Banashankari	Yes	Available
Duster	5	4700	HSR Layout	Yes	Available
i10	5	3000	Bellandur	Yes	Available
Fortuner	7	5000	Indiranagar	No	Available

15 rows in set (0.00 sec)

- 6) The user is then required to type the name of the car they want to rent, in the entry box provided at the bottom and click ‘SELECT’:



Please enter the name of the car you'd like to select:

- 7) This updates our database and changes the data in the ‘Availability’ column of the selected car, from ‘Available’ to ‘Unavailable’. It also displays a confirmation message on the window for the user:

carname	seater	price	pickup	driver	Availability
WagonR	5	2000	Shivajinagar	Yes	Available
Honda City	5	3500	Bellandur	Yes	Unavailable
Swift Dzire	5	3500	Koramangala	No	Available
Innova	7	4500	Koramangala	Yes	Available

- 8) If the ‘Return’ button is clicked instead of the ‘Rent’ button, a new window opens displaying options to specify the name of the car and the location at which the user wishes to return the car:



- 9) The user can select the car to be returned and possible drop-off locations by clicking on the drop-down menus, which display the following:



- 10) The database is then updated with the newly returned car and a confirmation message is displayed on the screen:

You returned a Jeep Compass at Koramangala

- 11) An entry box is present at the bottom of the window to fetch user feedback and improve services:



Conclusions & future enhancements

Bon Voyage Car Rentals provides services for easy car rentals and returns. Through this app, one can easily view and select from a wide range of cars, according to their preferences. The app only handles the process of data storage and updation with respect to renting or returning cars. It does not provide facilities for payment transactions or real-time verifications of rents and returns.

Features we'd like to include in future:

- Subscriptions for regular users, with options to sign-in and view previous rents/returns.
- Services to provide in-app payment transactions while renting/returning a vehicle.
- Improved features to allow the user to exercise multiple filters at once.
- Addition of a Google Maps feature to help the user navigate while using the rented car.
- Widen our database by including 2-wheelers and heavy duty vehicles.
- Include a range of electric vehicles to encourage environment-friendly activities.

References

- Python Tutorial - Master Python Programming For Beginners from Scratch. 2022. *Tkinter PhotoImage*. [online] Available at:
<<https://www.pythontutorial.net/tkinter/tkinter-photoimage/>> [Accessed 28 January 2022].
- Youtube.com. 2022. [online] Available at:
<https://www.youtube.com/watch?v=4IsLwwb_yDs&t=343s> [Accessed 28 January 2022].
- GeeksforGeeks. 2022. *Python | setting and retrieving values of Tkinter variable - GeeksforGeeks*. [online] Available at:
<<https://www.geeksforgeeks.org/python-setting-and-retrieving-values-of-tkinter-variable/>> [Accessed 28 January 2022].
- GeeksforGeeks. 2022. *Python Tkinter - Checkbutton Widget - GeeksforGeeks*. [online] Available at: <<https://www.geeksforgeeks.org/python-tkinter-checkbutton-widget/>> [Accessed 28 January 2022].
- Tutorialspoint.com. 2022. *Python - Tkinter place() Method*. [online] Available at:
<https://www.tutorialspoint.com/python/tk_place.htm> [Accessed 28 January 2022].
- Tkinter window, H., Gregory, M. and Gregory, M., 2022. *How do I close a tkinter window?*. [online] Stack Overflow. Available at:
<<https://stackoverflow.com/questions/110923/how-do-i-close-a-tkinter-window>> [Accessed 28 January 2022].
- Python Tutorial - Master Python Programming For Beginners from Scratch. 2022. *How to Use Tkinter Treeview To Display Tabular & Hierarchical Data*. [online] Available at:
<<https://www.pythontutorial.net/tkinter/tkinter-treeview/>> [Accessed 28 January 2022].
- W3schools.com. 2022. *Python MySQL*. [online] Available at:
<https://www.w3schools.com/python/python_mysql_getstarted.asp> [Accessed 28 January 2022].