

MINI - PROJECT REPORT ON

**“Car Recommendation System”**

BY

1. Aman Bharti (2193032)
2. Gaurav Kumar (2193104)
3. Chinmay Mandavkar (2193088)
4. Aman raj (2193033)

**Under the Guidance of**

***Prof. Shahin Makubhai***

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

**MIT SCHOOL of Engineering**

Loni Kalbhor Pune

**UNDERTAKING**

We declare that the work presented in this project titled “**Car Recommendation System**”, submitted to “**MIT ADT UNIVERSITY, SCHOOL OF ENGINEERING”** is our original work. Carried out from “August 2021 to December 2021” under the guidance of **Prof. Shahin Makubhai**.

**Group Members**

**Aman Bharti,**

**Gaurav Kumar,**

**Chinmay Mandavkar,**

**Aman raj**

**M.I.T. SCHOOL OF ENGINEERING**

DEPARTMENT OF COMPUTER ENGINEERING

LONI – KALBHOR PUNE

***CERTIFICATE***



This is to certify that the Mini- Project report entitled

“*Car Recommendation System* ”

**Submitted By**

Aman Bharti, Gaurav Kumar, Aman Raj, Chinmay Mandavkar

is a record of Bonafide work carried out by them, under my guidance, in partial fulfillment of the requirement for the Third Year of Engineering (Computer) at M.I.T. School of Engineering, Pune under MIT Art, Design & Technology University.

Date: 27/11/2021 Place: Pune

**Prof. Shahin Makubhai Dr. RajneeshKaur Sachdeo**

**Guide, Dean Engineering,**

**Department of CSE Head, Department of CSE**

**M.I.T. School of Engineering MIT School of Engineering**

**Loni Kalbhor, Pune Loni-Kalbhor, Pune**

**ACKNOWLEDGEMENT**

We take upon this opportunity to acknowledge the many people whose efforts and

support helped me complete this project. We would like to thank ourMini Project guide **Shahin Makubhai ma’am** for providing us this opportunity. We also express our deepest gratitude to our mentor **Prof. Shahin Makubhai** once again for providing such meaningful and efficient knowledge. Finally, I would like to wind up by paying my heartfelt thanks all my colleagues who

were there to help each other.

**Group Members**

**Aman Bharti,**

**Gaurav Kumar,**

**Chinmay Mandavkar,**

**Aman raj**

**CONTENTS**

* Acknowledgments
* Abstract
* Introduction
* Project Objective
* Requirements
* Project Purpose
* Overall description
* Modules
* Output Explanation
* Code
* Scope of the project
* Conclusion
* Reference

**ABSTRACT**

Having a vehicle has become a mandatory requirement in the modern world due to the complex lifestyle of the people. There are various types of domestic vehicles available in the market. Some of them are multi-purpose and some are used for specific purposes. Automobile industry invests a lot in producing different car models to cater the needs of their customers with different social and economic backgrounds. So, the prices of the vehicles with the same features varies in a wide range. Thus, choosing the most appropriate vehicle is a challenging task for the customers who are planning to buy a vehicle. It requires technical expertise to some extent and guidance to make the right choice by analyzing the customer’s requirements. Thus, most of the vehicle buyers seek services from experts or consultants before buying a vehicle. Therefore, we are investigating a potential solution for this issue using modern machine learning technologies.

**INTRODUCTION**

Choosing the most appropriate vehicle is a challenging task for the customers who are planning to buy a vehicle. It requires technical expertise to some extent and guidance to make the right choice by analyzing the customer’s requirements.

Thus, most of the vehicle buyers seek services from experts or consultants before buying a vehicle. Therefore, we are investigating a potential solution for this issue using modern machine learning technologies.

**PROJECT OBJECTIVE**

To develop a car recommendation system that will provide a platform for buyers to search the best car of his choice.

Users have to input the details like car price, fuel type etc.

**REQUIREMENTS**

### **1. User Interface Requirements**

Simple Graphical user interface (GUI) for easy navigating through the program

Easy to update profile and items

Dynamically configurable interface

Appealing to the eye through coloration and pictorial presentation

### 

### **2. Hardware Interface Requirements**

Processor speed of 0.5 GHz or more for mobile gadgets

Processor speed of 1.5ghz or more for desktop and computer gadgets

Ram of 500Mb and above for all devices

Free storage memory capacity of more than 100Mb

### 

### **3. Communication Interface Requirements**

1. No Internet - For Basic Comparison
2. Internet – For More Information

### 

### **4.Development tools**

1. Python
2. Pycharm IDE
3. Tkinter

## **PROJECT PURPOSE**

This system mainly helps clients without more knowledge about cars to purchase the car, also suggest more options available based on the customer requirement.

To provide accurate and up-to-date information to the user, links are provided for each car module.

## **OVERALL DESCRIPTION**

## 1. Product Perspective

The following are the main features;

Cross platform support – it offers operation support for most of the known and commercial operating systems including windows operating system.

2. Product Function

1. Enable the users to search for the appropriate cars according to their requirements.
2. Users can define a prerequisite fuel type according to their preference.
3. Users can get more personalized and detailed suggestions with OUR RECOMMENDATION option.

3. User Characteristics

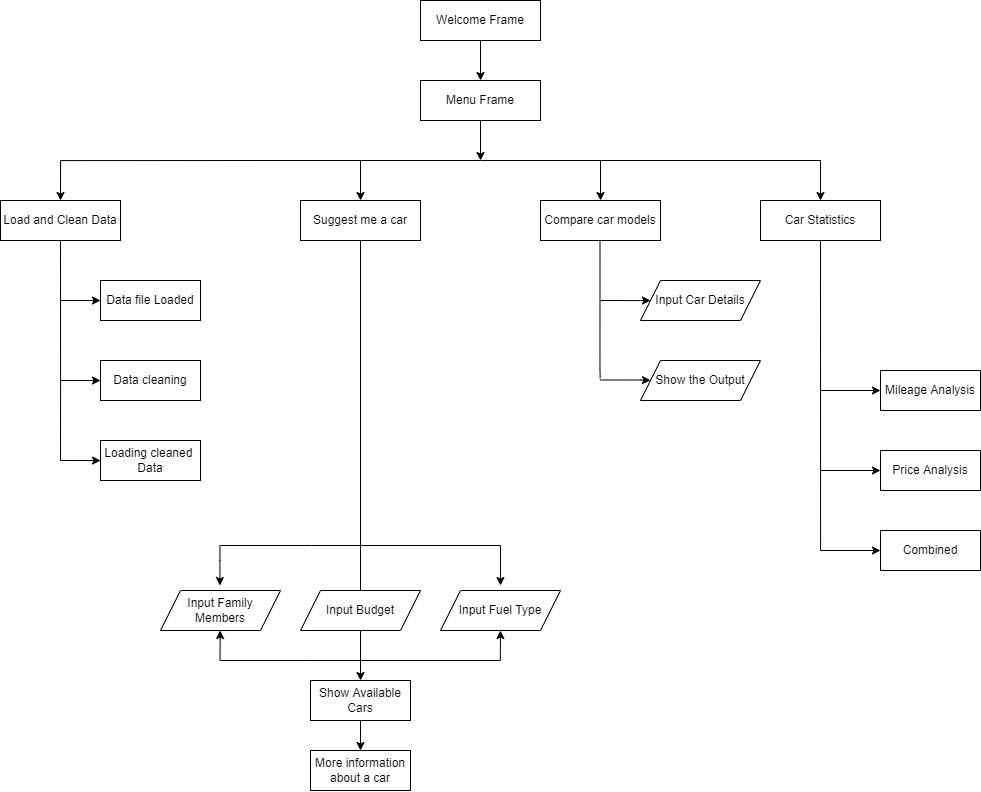
The system requires that the users be equipped with overall internet knowledge and the gadget accessing it. The administrator is expected to have more skills with the interface of the tech support system. The sellers should also have skills of well analyzing and navigating the internet usage and be able to handle customers who may not be so familiar with the internet.

4. Constraints

### The choice of the database to use should be considered carefully, since there will be a lot of data traffic and the large amounts of data will also judge the database. A good database will yield speed querying of data.

The device should be enabled with internet

5. System Flowchart



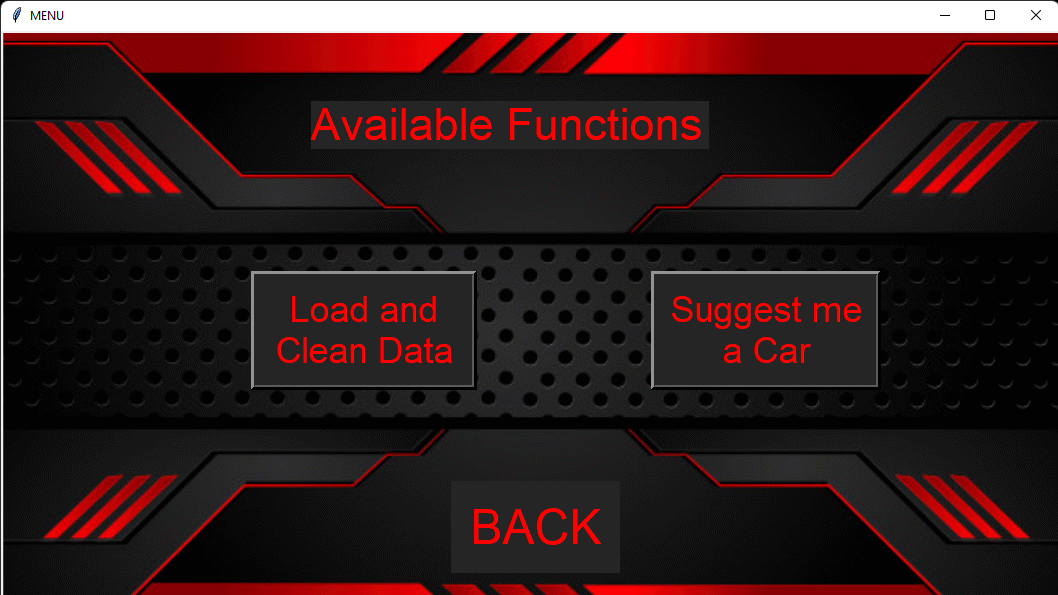
In this paper first module is home page, which consists of login screen for authenticate persons to entry and over view about the system. The second phase of projects consist of various available functions such as load and clean data, suggest me a car, Compare Car models and different statistics about various car modules. Each and every module has specific functions and operations.

**Modules**

**This is the first frame**

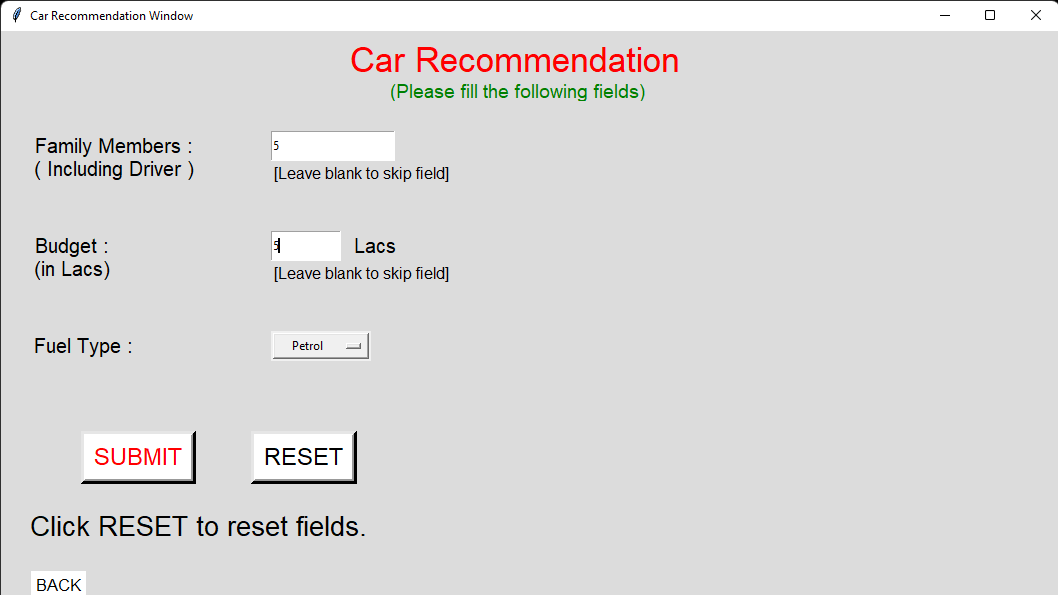
****

**This is the menu frame**

****

**Car Recommendation Window:**

Clicking on “Suggest me a Car” in the MENU FRAME brings you on this frame. This window looks as follows:

****

This window provides suggestion of car(s). User needs to input:

1. Number of family members

2. Budget (in Lacs)

3. Fuel Type

a. Petrol

b. Diesel

c. CNG

d. Anything (for flexibility)

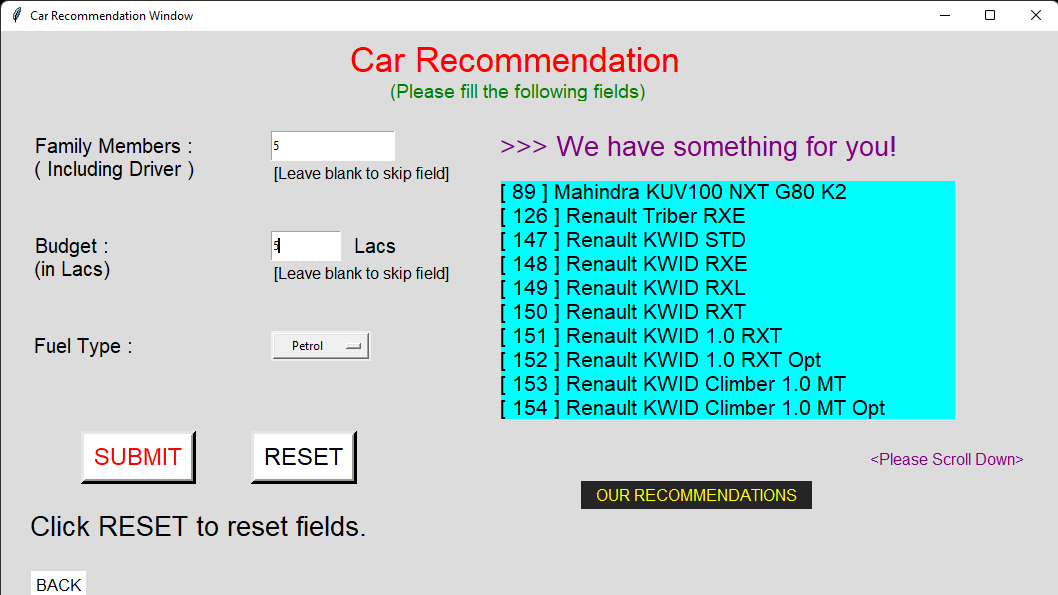
User can leave the field(s) BLANK to skip a field. This provides flexibility to the user. Test Cases:

1. If user enters a non-positive value, the system suggests to enter a value greater than ZERO. 2. If number of seats is beyond the limit, user is prompted for the same.

3. If user has less budget, he is suggested for a car loan.

4. If there are no cars in the dataset, user is notified for the same.

5. Input is in appropriate format.

****

Now, there are two buttons

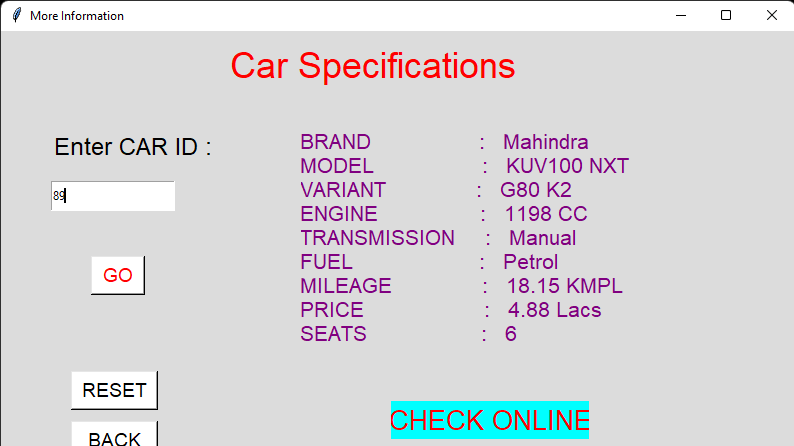
1. More Info

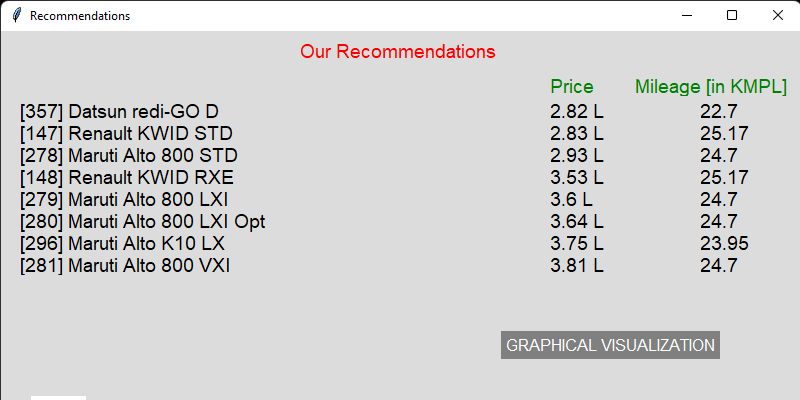
2. Our Recommendations

Button I: More Info

This button opens a window which gives car specifications to the user. The user needs to input the car ID which he/she is supposed to note in the previous frame (Suggestion Frame).

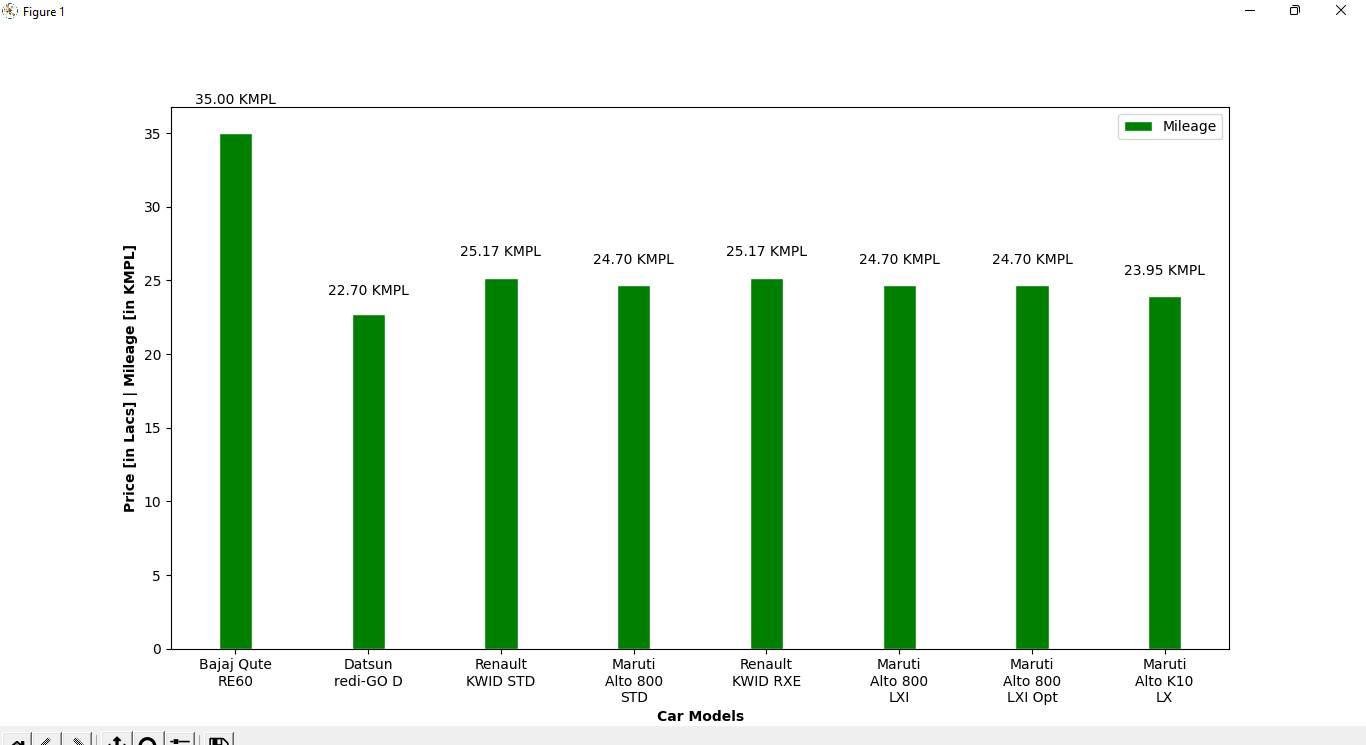
There is also an option to check online for a car model

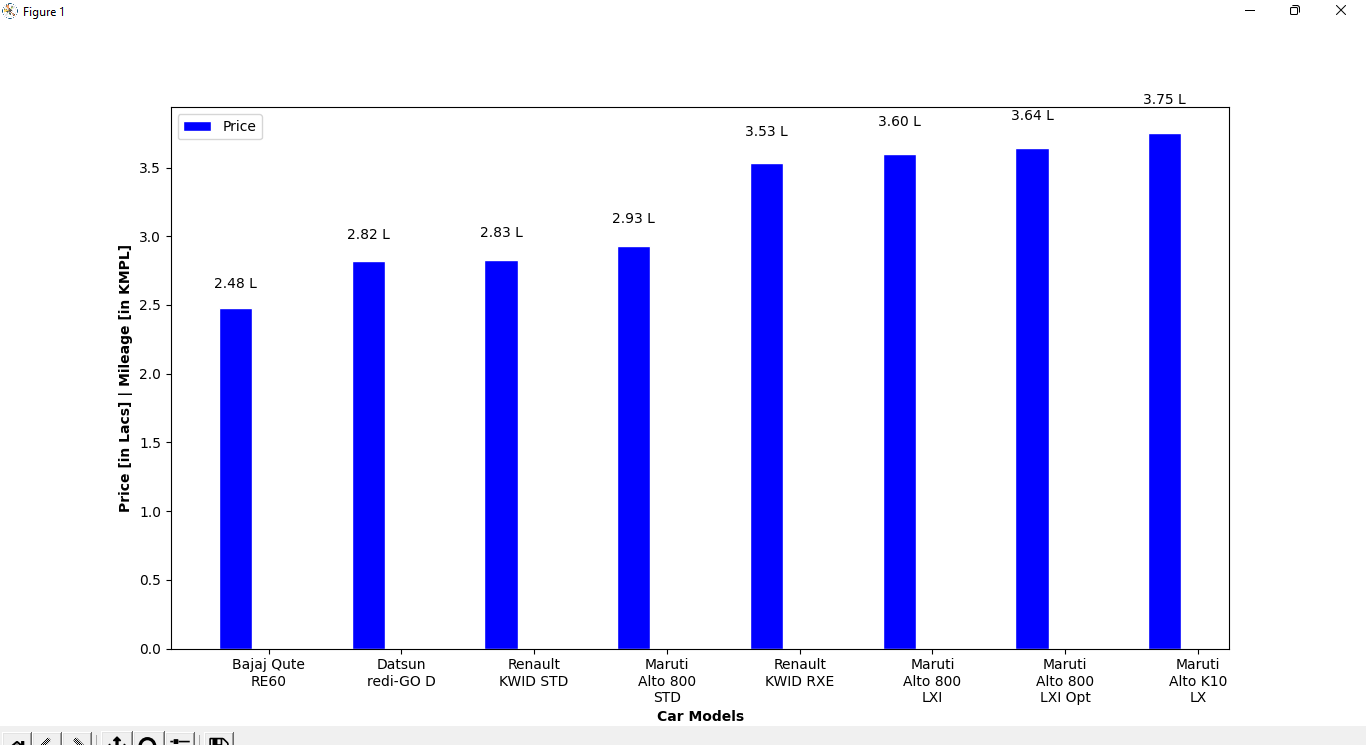
****

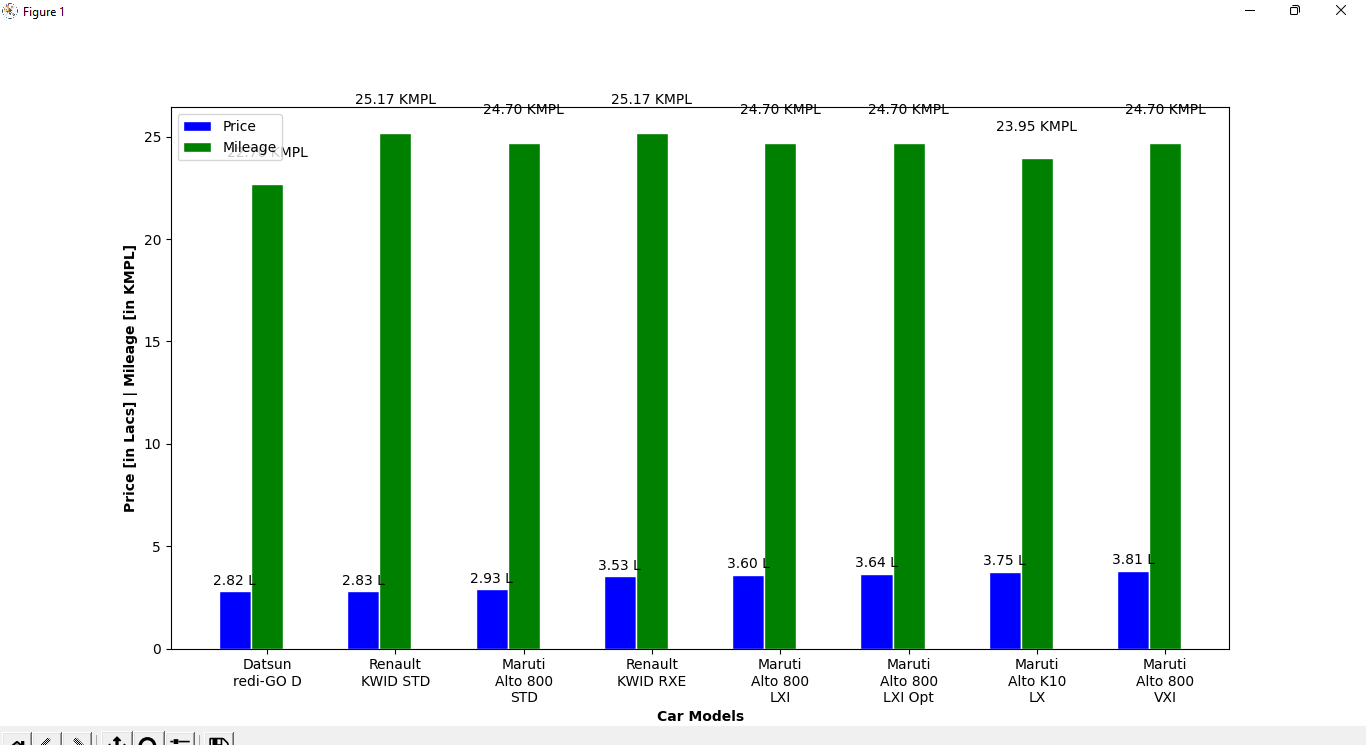
****

This window shows the Car Recommendations. This sorts the Cars according to user requirements, so that user gets the desired facilities in minimum price.

There is a buttons: Graphical Visualisation

****

****

****

**Codes**

**app.py**

from tkinter import \*  
import os  
  
frame1=Tk()  
  
def frame2():  
 os.system('python frame2.py')  
   
######################## Images  
image1 = PhotoImage(file = 'bg3.gif')  
background = Label(frame1, image=image1)  
background.place(x=0, y=0)  
#background.image = image1  
########################  
  
######################## Window  
frame1.title('Car Recommendation System')  
frame1.geometry('1280x720')  
windowWidth = 1058  
windowHeight = 595  
position\_x = int(frame1.winfo\_screenwidth()/2 - windowWidth/2)  
position\_y = int(frame1.winfo\_screenheight()/2 - windowHeight/2)  
frame1.geometry("+{}+{}".format(position\_x, position\_y))  
icon=Image('photo',file='icon1.png')  
frame1.tk.call('wm', 'iconphoto', frame1.\_w, icon)  
########################  
  
######################## Text Box  
text=Text(frame1, height=6, width=26, bg='black', fg='white', font=('',14), bd=-2)  
text.insert(5.0, "Under the Guidance of\nProf. Shahin Makubhai\nDEPARTMENT OF COMPUTER SCIENCE & ENGINEERING\nMIT SCHOOL of Engineering\nLoni Kalbhor Pune")  
text.config(state='disabled')  
text.place(x=10, y=50)  
########################  
  
######################## Buttons  
exit\_button = Button(frame1, text = 'EXIT', width = 6, command = frame1.destroy, fg='red', bg='#252525', font=('',30), bd=-2)  
exit\_button.place(x=150,y=545)  
  
continue\_button = Button(frame1, text = 'CONTINUE', width = 9, height = 1, bg='#7CFC00', fg='red', font=('',30), borderwidth=5, command=frame2)  
continue\_button.place(x=910,y=535)  
  
#b1\_button = Button(frame1, text = 'Developers Info', width = 12, height = 1, bg='white', fg='grey', font=('',24), borderwidth=5, command=f3)  
#b1\_button.place(x=970,y=85)  
########################  
  
frame1.mainloop()

**best\_mileage.py**

import matplotlib.pyplot as plt,csv  
from textwrap import wrap  
  
fo = open('data1\_clean.csv')  
data = csv.reader(fo,delimiter='|')  
headers = next(data)  
  
sort = sorted ( data,key=lambda t: float(t[7]), reverse=True )  
  
names=[]  
mileages=[]  
  
for row in sort:  
 names.append(row[1]+' '+row[2]+' '+row[3])  
 mileages.append(float(row[7]))  
   
if len(names)>8:  
  
 names=names[:8]  
 mileages=mileages[:8]  
  
names = [ '\n'.join(wrap(l, 10)) for l in names ]  
   
x = names  
x\_pos = [i for i, \_ in enumerate(x)]  
  
fig, ax = plt.subplots()  
rects1 = ax.bar(x\_pos, mileages, color='g', label='Mileage')  
plt.xlabel("Car Models")  
plt.ylabel("Mileage [in KMPL]")  
plt.title("Best Mileage Cars")  
plt.xticks(x\_pos, x)  
# Turning on the grid  
plt.minorticks\_on()  
plt.grid(which='major', linestyle='-', linewidth='0.5', color='red')  
# Customizing the minor grid  
plt.grid(which='minor', linestyle=':', linewidth='0.5', color='black')  
def autolabel(rects):  
 for rect in rects:  
 height = rect.get\_height()  
 ax.text(rect.get\_x() + rect.get\_width()/2., 1.05\*height,'%0.2f' % float(height),ha='center', va='bottom')  
autolabel(rects1)  
  
plt.legend()   
plt.show()

**best\_price.py**

import matplotlib.pyplot as plt,csv  
from textwrap import wrap  
  
fo = open('data1\_clean.csv')  
data = csv.reader(fo,delimiter='|')  
headers = next(data)  
  
sort = sorted ( data,key=lambda t: float(t[8]) )  
  
names=[]  
prices=[]  
for row in sort:  
 names.append(row[1]+' '+row[2]+' '+row[3])  
 prices.append(float(row[8]))  
   
if len(names)>8:  
 names=names[:8]  
 prices=prices[:8]  
  
names = [ '\n'.join(wrap(l, 10)) for l in names ]  
  
x = names  
x\_pos = [i for i, \_ in enumerate(x)]  
  
fig, ax = plt.subplots()  
rects1 = ax.bar(x\_pos, prices, color='cyan', label='Price')  
plt.xlabel("Car Models")  
plt.ylabel("Prices [in Lacs]")  
plt.title("Best Priced Cars")  
plt.xticks(x\_pos, x)  
  
def autolabel(rects):  
 for rect in rects:  
 height = rect.get\_height()  
 ax.text(rect.get\_x() + rect.get\_width()/2., 1.05\*height,'%0.2f L' % float(height),ha='center', va='bottom')  
autolabel(rects1)  
  
plt.legend()   
plt.show()

**best\_price\_mileage.py**

import matplotlib.pyplot as plt,csv,numpy as np  
from textwrap import wrap  
  
fo = open('data1\_clean.csv')  
data = csv.reader(fo,delimiter='|')  
headers = next(data)  
  
sort = sorted ( data,key=lambda t: float(t[8]) )  
  
names=[]  
prices=[]  
mileages=[]  
  
for row in sort:  
 names.append(row[1]+' '+row[2]+' '+row[3])  
 prices.append(float(row[8]))  
 mileages.append(float(row[7]))  
   
if len(names)>8:  
 names=names[:8]  
 prices=prices[:8]  
 mileages=mileages[:8]  
  
names = [ '\n'.join(wrap(l, 10)) for l in names ]  
  
barWidth = 0.25  
  
fig, ax = plt.subplots()  
# Set position of bar on X axis  
r1 = np.arange(len(names))  
r2 = [x + barWidth for x in r1]  
r3 = [x + barWidth for x in r2]  
   
# Make the plot  
rects1 = ax.bar(r1, prices, color='b', width=barWidth, edgecolor='white', label='Price')  
rects2 = ax.bar(r2, mileages, color='g', width=barWidth, edgecolor='white', label='Mileage')  
   
# Add xticks on the middle of the group bars  
plt.xlabel('Car Models', fontweight='bold')  
plt.ylabel('Price [in Lacs] | Mileage [in KMPL]', fontweight='bold')  
plt.xticks([r + barWidth for r in range(len(names))], names)  
plt.title('Best Priced Cars with Mileages')  
  
def autolabel1(rects):  
 for rect in rects:  
 height = rect.get\_height()  
 ax.text(rect.get\_x() + rect.get\_width()/2., 1.05\*height,'%0.2f L' % float(height),ha='center', va='bottom')  
autolabel1(rects1)  
def autolabel2(rects):  
 for rect in rects:  
 height = rect.get\_height()  
 ax.text(rect.get\_x() + rect.get\_width()/2., 1.05\*height,'%0.2f KMPL' % float(height),ha='center', va='bottom')  
autolabel2(rects2)  
   
# Create legend & Show graphic  
plt.legend()  
plt.show()

**frame2.py**

from tkinter import \*  
import os  
  
######################## Basic Structure  
frame2=Tk()  
frame2.geometry('1058x595')  
windowWidth = 1058  
windowHeight = 595  
position\_x = int(frame2.winfo\_screenwidth()/2 - windowWidth/2)  
position\_y = int(frame2.winfo\_screenheight()/2 - windowHeight/2)  
frame2.geometry("+{}+{}".format(position\_x, position\_y))  
frame2.title('MENU')  
########################  
  
######################## Functions  
def f1():  
 os.system('python load\_clean\_data.py')  
def f2():  
 os.system('python frame3.py')  
  
########################  
  
######################## Images  
image1 = PhotoImage(file = 'bg2.gif')  
background = Label(frame2, image=image1)  
background.place(x=0, y=0)  
#background.image = image1  
########################  
  
######################## Text Box  
text=Text(frame2, height=1, width=16, bg='#252525', fg='red', font=('',34), bd=-2)  
text.insert(5.0, "Available Functions")  
text.config(state='disabled')  
text.place(x=310, y=70)  
########################  
  
######################## Buttons  
back\_button = Button(frame2, text = 'BACK', command = frame2.destroy, font=('',36), bd=-2, fg='red', bg='#252525')  
back\_button.place(x=450,y=450)  
########################  
b1=Button(frame2, text='Load and\nClean Data', font=('',27),width=10, fg='red', bg='#252525', borderwidth=5, command=f1)  
b1.place(x=250,y=240)  
b2=Button(frame2, text='Suggest me\na Car', font=('',27), fg='red', bg='#252525', borderwidth=5, command=f2)  
b2.place(x=650,y=240)  
  
frame2.mainloop()

**frame3.py**

from tkinter import \*  
import os  
  
######################## Window  
frame3=Tk()  
frame3.title('Car Recommendation Window')  
frame3.config(background='#DCDCDC')  
frame3.geometry('1058x595')  
windowWidth = 1058  
windowHeight = 595  
position\_x = int(frame3.winfo\_screenwidth()/2 - windowWidth/2)  
position\_y = int(frame3.winfo\_screenheight()/2 - windowHeight/2)  
frame3.geometry("+{}+{}".format(position\_x, position\_y))  
########################  
######################## Function Definitions  
def reset\_frame():  
 frame3.destroy()  
 os.system('python frame3.py')  
   
def more\_info():  
 os.system('python more\_info.py')  
  
def car\_loans():  
 os.system('python car\_loans.py')  
   
def recommendations():  
 os.system('python recommendations.py')  
   
def suggest():  
 if entry1.get() == '':  
 family\_members = 1  
 else:  
 family\_members = int(entry1.get())  
 if entry2.get() == '':  
 budget = 8000  
 else:  
 budget = float(entry2.get())  
 fuel = fuel\_type.get()  
  
 import csv  
 fo = open('data1\_clean.csv')  
 data = csv.reader(fo, delimiter = '|')  
   
 fo1 = open('temp.csv','w')  
 fo1.write('ID|BRAND|MODEL|VARIANT|ENGINE|TRANSMISSION|FUEL|MILEAGE|PRICE|SEATS|URL')  
  
 ######################## Minimum Price of Car  
 fo.seek(0)  
 headers = next(data)  
 min\_price = 8000  
 for row in data:  
 if min\_price >= float(row[8]):  
 min\_price = float(row[8])  
 ######################## Maximum Number of Seats   
 fo.seek(0)  
 headers = next(data)  
 max\_seats = 0  
 for row in data:  
 if max\_seats <= int(row[9]):  
 max\_seats = int(row[9])  
 ########################   
   
 if family\_members<=0 or budget<=0:  
 err\_text=Text(frame3, height=5, width=27, bg='#DCDCDC', font=('',20), bd=-2)  
 err\_text.insert(5.0, "Please enter a value greater than ZERO...\n\nClick RESET button to retry.")  
 err\_text.config(state='disabled')  
 err\_text.place(x=500, y=250)  
 else:  
 if family\_members>max\_seats:  
 err\_text=Text(frame3, height=8, width=32, bg='#DCDCDC', font=('',20), bd=-2)  
 err\_text.insert(5.0, "Dear User, a car has got limited number of seats.\n\nPlease consider buying some other \nmotor vehicle.\n\nClick RESET button to retry.")  
 err\_text.config(state='disabled')  
 err\_text.place(x=500, y=200)  
   
 else:  
 if budget<min\_price:  
 err\_text=Text(frame3, height=8, width=30, bg='#DCDCDC', font=('',20), bd=-2)  
 err\_text.insert(5.0, "Sorry, we don't have any CAR MODEL in this price segment.\nPlease try to increase your budget.\n\nGOOD NEWS :\nGood schemes are available on \nCAR LOANS.\nCheck them out! See you soon :-)")  
 err\_text.config(state='disabled')  
 err\_text.place(x=500, y=150)  
   
 loan\_button = Button(frame3, text = 'CHECK CAR LOANS', width = 15, fg='yellow', bg='#252525', font=('',14), bd=-2, command=car\_loans)  
 loan\_button.place(x=630,y=450)  
   
 else:  
 count=0  
 if fuel=='Anything' :  
 fo.seek(0)  
 headers = next(data)  
 for row in data:  
 if float(row[8])<=budget and int(row[9])>=family\_members:  
 count=count+1  
 if count>0:  
 fo.seek(0)  
 headers = next(data)  
 str\_result=''  
 for row in data:  
 if float(row[8])<=budget and int(row[9])>=family\_members:  
 str\_result = str\_result+'[ '+row[0]+' ] '+row[1]+' '+row[2]+' '+row[3]+'\n'  
   
 str\_temp='|'.join(row)  
 fo1.write('\n')  
 fo1.write(str\_temp)  
 fo1.close()  
  
 else:  
 fo.seek(0)  
 headers = next(data)  
 for row in data:  
 if row[6]==fuel and float(row[8])<=budget and int(row[9])>=family\_members:  
 count=count+1  
 if count>0:  
 fo.seek(0)  
 headers = next(data)  
 str\_result=''  
 for row in data:  
 if row[6]==fuel and float(row[8])<=budget and int(row[9])>=family\_members:  
 str\_result = str\_result+'[ '+row[0]+' ] '+row[1]+' '+row[2]+' '+row[3]+'\n'  
   
 str\_temp='|'.join(row)  
 fo1.write('\n')  
 fo1.write(str\_temp)  
 fo1.close()  
   
 if count>0:  
 text3=Text(frame3, height=1, width=30, bg='#DCDCDC',fg='purple', font=('',20), bd=-2)  
 text3.insert(5.0, ">>> We have something for you!")  
 text3.config(state='disabled')  
 text3.place(x=500, y=100)  
   
 text\_result = Text(frame3, height=10, width=38, bg='cyan', font=('',16), bd=-2)  
 text\_result.insert(5.0, str\_result)  
 text\_result.config(state='disabled')  
 text\_result.place(x=500, y=150)  
 recommendations\_button = Button(frame3, text = 'OUR RECOMMENDATIONS', width = 25, fg='yellow', bg='#252525', font=('',12), bd=-2, command=recommendations)  
 recommendations\_button.place(x=580,y=450)  
   
 else:  
 err\_text=Text(frame3, height=8, width=27, bg='#DCDCDC', font=('',20), bd=-2)  
 err\_text.insert(5.0, "There are currently NO CARS \nmeeting your requirements...\nWe hope to see you back soon.\n\nClick RESET button to modify your requirements.")  
 err\_text.config(state='disabled')  
 err\_text.place(x=500, y=200)  
   
 if count>10:  
 text3=Text(frame3, height=1, width=18, bg='#DCDCDC',fg='purple', font=('',12), bd=-2)  
 text3.insert(5.0, "<Please Scroll Down>")  
 text3.config(state='disabled')  
 text3.place(x=870, y=420)  
   
######################## Text Box  
text1=Text(frame3, height=1, width=18, bg='#DCDCDC',fg='red', font=('',26), bd=-2)  
text1.insert(5.0, "Car Recommendation")  
text1.config(state='disabled')  
text1.place(x=350, y=10)  
  
text1=Text(frame3, height=1, width=25, bg='#DCDCDC',fg='green', font=('',14), bd=-2)  
text1.insert(5.0, "(Please fill the following fields)")  
text1.config(state='disabled')  
text1.place(x=390, y=50)  
  
text2=Text(frame3, height=1, width=25, bg='#DCDCDC', font=('',20), bd=-2)  
text2.insert(5.0, "Click RESET to reset fields.")  
text2.config(state='disabled')  
text2.place(x=30, y=480)  
########################  
######################## Fields  
label1 = Label(frame3, text='Family Members :\n( Including Driver )', font=('',15), bg='#DCDCDC')  
label1.place(x=30,y=100)  
entry1=Entry(frame3)  
entry1.place(x=270,y=100,height=30)  
temp1=Label(frame3, text='[Leave blank to skip field]', font=('',12), bg='#DCDCDC')  
temp1.place(x=270,y=130)  
   
label2 = Label(frame3, text='Budget :\n(in Lacs)', font=('',15), bg='#DCDCDC')  
label2.place(x=30,y=200)  
entry2=Entry(frame3)  
entry2.place(x=270,y=200,height=30,width=70)  
lacs=Label(frame3, text='Lacs', font=('',15), bg='#DCDCDC')  
lacs.place(x=350,y=200)  
temp2=Label(frame3, text='[Leave blank to skip field]', font=('',12), bg='#DCDCDC')  
temp2.place(x=270,y=230)  
  
fuel\_type=StringVar()  
fuel\_type.set('Petrol')  
label3 = Label(frame3, text='Fuel Type :', font=('',15), bg='#DCDCDC')  
label3.place(x=30,y=300)  
entry3=OptionMenu(frame3,fuel\_type,'Petrol','Diesel','CNG','Anything')  
entry3.place(x=270,y=300,height=30,width=100)  
########################  
submit\_button=Button(frame3, text = 'SUBMIT', font=('',18), fg='red', bg='white', borderwidth=5, command=suggest)  
submit\_button.place(x=80,y=400)  
reset\_button=Button(frame3, text = 'RESET', font=('',18), bg='white', borderwidth=5, command=reset\_frame)  
reset\_button.place(x=250,y=400)  
back\_button = Button(frame3, text = 'BACK', command = frame3.destroy, font=('',12), bd=-2, bg='white')  
back\_button.place(x=30,y=540)  
back\_button = Button(frame3, text = 'MORE INFO', command = more\_info, font=('',12), bd=-2, bg='white')  
back\_button.place(x=120,y=540)  
########################  
frame3.mainloop()

**load\_clean\_data.py**

import csv, os  
from tkinter import \*  
  
######################## Loading Dataset into Memory  
file\_object = open('data1.csv')  
dataset = csv.reader(file\_object, delimiter='|')  
headers = next(dataset)  
######################## Dataset Loaded  
  
  
######################## Opening new file for Data Cleaning  
fo=open('data1\_clean.csv','w')  
######################## Writing Header  
fo.write('ID|BRAND|MODEL|VARIANT|ENGINE|TRANSMISSION|FUEL|MILEAGE|PRICE|SEATS|URL')  
  
######################## Cleaning Data  
for row in dataset:  
   
 row[4]=row[4].replace(' cc','')  
 row[4]=row[4].replace('cc','')  
 row[4]=row[4].replace('cc ','')  
 row[4]=row[4].replace(' cc ','')  
   
 row[7]=row[7].replace(' kmpl','')  
 row[7]=row[7].replace('kmpl','')  
 row[7]=row[7].replace('kmpl ','')  
 row[7]=row[7].replace(' kmpl ','')  
   
 row[8]=row[8].replace(' Rs.','')  
 row[8]=row[8].replace('Rs.','')  
 row[8]=row[8].replace('Rs. ','')  
 row[8]=row[8].replace(' Rs. ','')  
   
 str1='|'.join(row)  
 fo.write('\n')  
 fo.write(str1)  
   
fo.close()  
######################## Data Cleaned  
  
######################## Show cleaned data  
def show\_data():  
 os.system("python3 show\_data.py")  
  
######################## Acknowledgement  
ack=Tk()  
ack.title('Acknowledgement')  
ack.geometry('450x150')  
windowWidth = 450  
windowHeight = 150  
position\_x = int(ack.winfo\_screenwidth()/2 - windowWidth/2)  
position\_y = int(ack.winfo\_screenheight()/2 - windowHeight/2)  
ack.geometry("+{}+{}".format(position\_x, position\_y))  
########################  
text=Text(ack, bg='#DCDCDC', bd=-2, font=('',14))  
text.insert(5.0,'\n Data Loaded and Cleaned Successfully!')  
text.config(state='disabled')  
text.pack()  
ok\_button = Button(ack, text = 'OK', command = ack.destroy, width = 4, font=('',16), bd=-2, fg='yellow', bg='#252525')  
ok\_button.place(x=85,y=100)  
show\_data = Button(ack, text = 'Show Data', command = show\_data, width = 9, font=('',16), bd=-2, fg='yellow', bg='#252525')  
show\_data.place(x=205,y=100)  
ack.mainloop()  
######################## Data Loaded, Cleaned, Acknowledged to User

**more\_info.py**

from tkinter import \*  
import csv, os, webbrowser  
fo = open('data1\_clean.csv')  
data = csv.reader(fo, delimiter = '|')  
  
######################## Basic Structure  
root=Tk()  
root.title('More Information')  
root.geometry('794x446')  
windowWidth = 794  
windowHeight = 446  
position\_x = int(root.winfo\_screenwidth()/2 - windowWidth/2)  
position\_y = int(root.winfo\_screenheight()/2 - windowHeight/2)  
root.geometry("+{}+{}".format(position\_x, position\_y))  
root.config(background='#DCDCDC')  
  
######################## Maximum Car ID  
fo.seek(0)  
headers = next(data)  
max\_id = 0  
for row in data:  
 if max\_id <= int(row[0]):  
 max\_id = int(row[0])  
########################  
  
######################## Function Definition  
def buy\_now(url):  
 webbrowser.open(url)  
  
def reset\_frame():  
 root.destroy()  
 os.system('python3 more\_info.py')  
   
def show\_info():  
 identifier = int(entry1.get())  
 if identifier<=0 or identifier>max\_id:  
 err\_text=Text(root, height=5, width=27, bg='#DCDCDC', font=('',20), bd=-2)  
 err\_text.insert(5.0, "Please enter a valid CAR ID...\n\nClick RESET button to retry.")  
 err\_text.config(state='disabled')  
 err\_text.place(x=350, y=150)  
   
 else:  
 fo.seek(0)  
 headers = next(data)  
 for row in data:  
 if identifier==int(row[0]):  
 str1='BRAND : '+row[1]+'\nMODEL : '+row[2]+'\nVARIANT : '+row[3]+'\nENGINE : '+row[4]+' CC'+'\nTRANSMISSION : '+row[5]+'\nFUEL : '+row[6]+'\nMILEAGE : '+row[7]+' KMPL'+'\nPRICE : '+row[8]+' Lacs'+'\nSEATS : '+row[9]  
 url = row[10]  
 text1 = Text(root, height=10, width=35, bg='#DCDCDC',fg='purple', font=('',16), bd=-2)  
 text1.insert(5.0, str1)  
 text1.config(state='disabled')  
 text1.place(x=300, y=100)  
   
 link1 = Label(root, text='CHECK ONLINE',fg='red',bg='cyan',width=12,cursor='hand2',font=('',20))  
 link1.place(x=390,y=370)  
 link1.bind('<Button-1>',lambda e:buy\_now(url))  
   
  
######################## Text Box  
text1=Text(root, height=1, width=16, bg='#DCDCDC',fg='red', font=('',27), bd=-2)  
text1.insert(5.0, "Car Specifications")  
text1.config(state='disabled')  
text1.place(x=230, y=15)  
  
label1 = Label(root, text='Enter CAR ID :', font=('',18), bg='#DCDCDC')  
label1.place(x=50,y=100)  
entry1=Entry(root)  
entry1.place(x=50,y=150,height=30)  
  
######################## Button  
go\_button=Button(root, text = 'GO',width=4, font=('',14), fg='red', bg='white', command=show\_info)  
go\_button.place(x=90,y=225)  
reset\_button=Button(root, text = 'RESET',width=7, font=('',14), bg='white', command=reset\_frame)  
reset\_button.place(x=70,y=340)  
back\_button = Button(root, text = 'BACK',width=7, command = root.destroy, font=('',14), bg='white')  
back\_button.place(x=70,y=390)  
  
root.mainloop()

**recommendation\_graph.py**

import matplotlib.pyplot as plt,csv,numpy as np  
from textwrap import wrap  
  
fo\_temp = open('temp.csv')  
data\_temp = csv.reader(fo\_temp,delimiter='|')  
headers = next(data\_temp)  
  
sort = sorted ( data\_temp,key=lambda t: float(t[8]) )  
  
names=[]  
prices=[]  
mileages=[]  
  
for row in sort:  
 names.append(row[1]+' '+row[2]+' '+row[3])  
 prices.append(float(row[8]))  
 mileages.append(float(row[7]))  
   
if len(names)>8:  
 names=names[:8]  
 prices=prices[:8]  
 mileages=mileages[:8]  
  
names = [ '\n'.join(wrap(l, 10)) for l in names ]  
  
barWidth = 0.25  
  
fig, ax = plt.subplots()  
# Set position of bar on X axis  
r1 = np.arange(len(names))  
r2 = [x + barWidth for x in r1]  
r3 = [x + barWidth for x in r2]  
   
# Make the plot  
rects1 = ax.bar(r1, prices, color='b', width=barWidth, edgecolor='white', label='Price')  
rects2 = ax.bar(r2, mileages, color='g', width=barWidth, edgecolor='white', label='Mileage')  
   
# Add xticks on the middle of the group bars  
plt.xlabel('Car Models', fontweight='bold')  
plt.ylabel('Price [in Lacs] | Mileage [in KMPL]', fontweight='bold')  
plt.xticks([r + barWidth for r in range(len(names))], names)  
  
def autolabel1(rects):  
 for rect in rects:  
 height = rect.get\_height()  
 ax.text(rect.get\_x() + rect.get\_width()/2., 1.05\*height,'%0.2f L' % float(height),ha='center', va='bottom')  
autolabel1(rects1)  
def autolabel2(rects):  
 for rect in rects:  
 height = rect.get\_height()  
 ax.text(rect.get\_x() + rect.get\_width()/2., 1.05\*height,'%0.2f KMPL' % float(height),ha='center', va='bottom')  
autolabel2(rects2)  
   
# Create legend & Show graphic  
plt.legend()  
plt.show()

**recommendation.py**

import csv, os  
from tkinter import \*  
  
fo\_temp = open('temp.csv')  
data\_temp = csv.reader(fo\_temp,delimiter='|')  
headers = next(data\_temp)  
  
  
sort = sorted ( data\_temp,key=lambda t: float(t[8]) )  
  
str\_models=''  
str\_prices=''  
str\_mileages=''  
for row in sort:  
 str\_models=str\_models+'['+row[0]+']'+' '+row[1]+' '+row[2]+' '+row[3]+'\n'  
 str\_prices=str\_prices+row[8]+' L'+'\n'  
 str\_mileages=str\_mileages+row[7]+'\n'  
   
  
root=Tk()  
root.title('Recommendations')  
root.geometry('800x400')  
windowWidth = 800  
windowHeight = 400  
position\_x = int(root.winfo\_screenwidth()/2 - windowWidth/2)  
position\_y = int(root.winfo\_screenheight()/2 - windowHeight/2)  
root.geometry("+{}+{}".format(position\_x, position\_y))  
root.config(bg='#DCDCDC')  
  
def more\_info():  
 os.system('python more\_info.py')  
   
def graphical\_visualization():  
 os.system('python recommendation\_graph.py')  
  
  
text=Text(root, height=1, width=25, bg='#DCDCDC', fg='red', font=('',14), bd=-2)  
text.insert(5.0, "Our Recommendations")  
text.config(state='disabled')  
text.place(x=300, y=10)  
  
  
text=Text(root, height=8, width=100, bg='#DCDCDC', fg='black', font=('',14), bd=-2)  
text.insert(5.0, str\_models)  
text.config(state='disabled')  
text.place(x=20, y=70)  
  
text=Text(root, height=8, width=50, bg='#DCDCDC', fg='black', font=('',14), bd=-2)  
text.insert(5.0, str\_prices)  
text.config(state='disabled')  
text.place(x=550, y=70)  
  
text=Text(root, height=8, width=50, bg='#DCDCDC', fg='black', font=('',14), bd=-2)  
text.insert(5.0, str\_mileages)  
text.config(state='disabled')  
text.place(x=700, y=70)  
  
text=Text(root, height=1, width=25, bg='#DCDCDC', fg='green', font=('',14), bd=-2)  
text.insert(5.0, "Price")  
text.config(state='disabled')  
text.place(x=550, y=45)  
  
text=Text(root, height=1, width=25, bg='#DCDCDC', fg='green', font=('',14), bd=-2)  
text.insert(5.0, "Mileage [in KMPL]")  
text.config(state='disabled')  
text.place(x=635, y=45)  
  
back\_button = Button(root, text = 'BACK', command = root.destroy, font=('',12), bd=-2, bg='white')  
back\_button.place(x=30,y=365)  
  
graph\_button = Button(root, text = 'GRAPHICAL VISUALIZATION', command = graphical\_visualization, font=('',12), bd=-2, bg='grey',fg='white')  
graph\_button.place(x=500,y=300)  
   
root.mainloop()

**show\_data.py**

import csv, os  
from tkinter import \*  
  
######################## Loading Dataset into Memory  
file\_object = open('data1\_clean.csv')  
dataset = csv.reader(file\_object, delimiter='|')  
headers = next(dataset)  
str1 = "ID\tBRAND\tMODEL\t\tVARIANT\t\tENGINE\tTRANSMISSION\tFUEL\tMILEAGE\tPRICE\tSEATS"  
str1 += "\n----------------------------------------------------------------------------------------------------------------------------------------------------------"  
  
for row in dataset:  
 str1 += "\n"+row[0]+"\t"+row[1]+"\t"+row[2]+"\t\t"+row[3]+"\t\t"+row[4]+"\t"+row[5]+"\t"+row[6]+"\t"+row[7]+"\t"+row[8]+"\t"+row[9]  
  
######################## GUI  
data = Tk()  
data.title("Cleaned Data")  
data.geometry("1058x595")  
windowWidth = 1058  
windowHeight = 595  
position\_x = int(data.winfo\_screenwidth()/2 - windowWidth/2)  
position\_y = int(data.winfo\_screenheight()/2 - windowHeight/2)  
data.geometry("+{}+{}".format(position\_x, position\_y))  
data.config(bg='#DCDCDC')  
  
text=Text(data, width=100, bg='#DCDCDC', fg='black', font=('',12), bd=-2)  
text.insert(5.0, str1)  
text.config(state='disabled')  
text.place(x=60, y=10)  
  
back\_button = Button(data, text = 'BACK', command = data.destroy, width = 6, font=('',16), bd=-2, fg='yellow', bg='#252525')  
back\_button.place(x=480,y=500)  
  
data.mainloop()

**Future Scope**

This system can be made online, so than it can be accessed by the user from anywhere anytime. More specifications like availability of back camera, sunroof, number of airbags, different types of sensers, ground clearance and many more can be added for the user to choose from.

**Conclusion**

This system is mainly help clients to purchase car without more knowledge about car, also suggests various options available based on the customer requirements. This system has the potential to reduce a lot of research work of the user. It can correctly suggest the user car model(s) based on his/her requirements. To provide accurate and up-to-date information to the user, links are provided for each car model. The user can click on the link and thereby get the latest, updated information on the internet. This is very useful as the user can view the images, 360 degrees view of the car, reviews of other users, ratings, etc. Data Visualization helps to see the analysis of the available car models at a glance.

**Reference**

* YouTube
* GitHub
* **Prof. Sahin Makubhai**
* Google

**THANK YOU**