PyCode Project Report

Vehicle Safety System, Emergency Telemetrics and Visualization

by

Vraj Parikh, Roll number: 08 (TE EXTC B)

Abhishek Rajput, Roll number: 18(TE EXTC B)

Manav Parekh: 07(TE EXTC B)

Shreya Kini, Roll number: 43(TE EXTC A)

TE EXTC A/B

Submitted in partial fulfillment for

PyCode: Programming Course

organized by

Dept. Electronics and Telecommunication Engg., SFIT

AY: 2020-21

Table of Contents

I. Introduction of Topic

India is one of the busiest countries in the world in terms of road traffic. The automotive industry across the south Asian country became the fourth largest in the world in 2017. In 2019, there were almost three million new car registrations in the country.

The Indian road network, spanning over five million kilometers, carried almost 90 percent of the country's passenger traffic and about 65 percent of the goods. With the rapid increase in the number of cars and the mercilessly congested Indian roads, road safety has turned into a factor of utmost importance for the country's citizens.

In 2018, there were around 151 thousand deaths due to road accidents in India. One of the contributing factors could be the ever-increasing vehicle population.

II. Problem and Solution

- **Speeding:** Major factor contributing to the increased number of road accidents is speeding. The public fails to follow the speed limits, especially on the highway. This has resulted in 41% of the total deaths due to road accidents in India in 2014.

- **Drunken driving:** Even though driving under the influence of alcohol is strictly prohibited, many flaunt this rule, which at times results in road accidents. Even if the person under the influence of alcohol walks away safe from the scene of the accident due to the safety features of the car, the pedestrians and smaller vehicles involved in the accidents are not so lucky.

-

SOLUTION:

The **Internet of things** (**IoT**) is a system of interrelated computing devices, mechanical and digital machines provided with unique identifiers (UIDs) and the ability to transfer data over a network without requiring human-to-human or human-to-computer interaction.

IoT empowers persons and objects in the **transportation** system, helping them make informed and automated decisions to improve traffic flows: Commuters can better decide which route to choose, when to travel, when to take public **transport** instead of a car.

With more and more cities investing in smart infrastructures, IoT-powered vehicles are much better prepared to help drivers in commuting safely.

III. Source Code

```
'''



=============== This Code is Developed by Vraj Parikh
and Team================

* Vraj Parikh   T.E EXTC - B - 08
* Abhishek Rajput T.E EXTC - B - 18
* Shreya kini T.E EXTC - A - 43
* Manav ParekhT.E EXTC - B - 07

==========================Instructions For Successfull
y Running This Code============
1.Open Command Prompt(Windows+R)
2.Type python --version
>> 3.x.x

3. To install the required Library :
    pip install __________

4.Install MySql workbench , shell , server (Developer
Defaults).

5.You don't need to create a database , we have given
the command for the same in \
  the code.

6.Connect arduino , check your COM Port.
Open Device Manager, and expand the Ports (COM & LPT)
list.
Note the number on the USB Serial Port.
```

```
and set the BAUD RATE AS 9600.

7.Burn the code 'BLDC.ino' into arduino UNO/Nano.

8.And finally please don't register with FAKE Contact
Number because the message is  Very Critical\
    and may lead to certain issues. So Please Register
 using your Genuine Account Number.

'''
```

```
#IMPORTING THE LIBRARIES FROM PYTHON
from tkinter import messagebox
from tkinter import *
from tkinter import ttk
#from PIL import image
import mysql.connector
from  charging_vs_discharging import *
from randspeed import *
from arduino import *
#from sms import *
```

```
import requests
import json
import time
from tkinter.messagebox import showinfo,showerror
import datetime
import urllib.request
import requests
import folium
from gtts import gTTS
from playsound  import playsound

def createDatabase():

    mydb=mysql.connector.connect(

        host='localhost',
        user='root',
        passwd='bmwm53552',
        #database='pycode'

        )
    myCursor= mydb.cursor()
    myCursor.execute('CREATE DATABASE IF NOT EXISTS py
code')

    mydb.commit()
    mydb.close()

createDatabase()

def createTable():
    mydb=mysql.connector.connect(

        host='localhost',
```

```
        user='root',
        passwd='bmwm53552',
        database='pycode'

        )
    myCursor= mydb.cursor()




    myCursor.execute('CREATE TABLE IF NOT EXISTS usser
egisteration (Name VARCHAR(45),VIN VARCHAR(45) \
        ,Blood_Group VARCHAR(45),Contact_Number VARCHA
R(45),Emergency_Number VARCHAR(45) \
            ,State VARCHAR(45) ,City VARCHAR(45) ,Pinc
ode VARCHAR(45))' )


    mydb.commit()
    mydb.close()

createTable()



def send_sms(number,message):
    url='https://www.fast2sms.com/dev/bulk'
    params={
        'authorization':'itSyDVX57cgsvdTb1WqL9QEamH486
Pkz3RA0rhNopUMYx2BeIGvIjoUelRLYg61xWhuVnaQNXkcyBzqf',
        'sender_id':'FSTSMS',
        'message':message,
        'language':'English',
```

```
        'route':'p',
        'numbers':number


    }

    response=requests.get(url, params=params)
    dic=response.json()
    print(dic)
    return dic.get('return')
#=====================================Maps start=========
====================

res=requests.get('https://ipinfo.io/')
data = res.json()


#print(data)

location = data['loc'].split(',')
latitude=float(location[0])
longitude=float(location[1])
pincode = data['postal']
place=data['city'].split(',')
print(f'Co-
ordinates are latitude={latitude} , longitude={longitu
de}')
print(f'city={place} and pincode = {pincode}')

fg = folium.FeatureGroup('my map')
fg.add_child(folium.GeoJson(data=(open('india_states.j
son','r',encoding='UTF-8-sig').read())))
```

```
fg.add_child(folium.Marker(location=[latitude,longitud
e],popup='this is the site of emergency'))

map=folium.Map(location=[latitude,longitude],zoom_star
t=10)
map.add_child(fg)


map.save("locate.html")
print('file:///C:/Users/Vrrajjj%20M5/Desktop/python/sm
s%20project/locate.html')

#==================Maps end============================
=




#Creating Root class for TKINTER LIBRARY


window = Tk()
window.title("Vehicle Configuation")
window.geometry('400x300')
window.configure(background = "#275296")





#========== CREATING LABELS ==============

#Label Widgets
```

```
userNameLabel = Label(window ,text = "Full Name")
userNameLabel.grid(row = 0,column = 3)

vehicleNumberLabel = Label(window ,text = "Vehicle Num
ber")
vehicleNumberLabel.grid(row = 2,column = 3)

bloodGroupLabel = Label(window ,text = "Blood Group")
bloodGroupLabel.grid(row = 4,column = 3)

contactNumberLabel = Label(window ,text = "Contact Num
ber")
contactNumberLabel.grid(row = 6,column = 3)

emergencyNumberLabel =  Label(window ,text = "Emergenc
y Number")
emergencyNumberLabel.grid(row = 8,column = 3)

stateLabel =  Label(window ,text = "State")
stateLabel.grid(row = 10,column = 3)

cityLabel =  Label(window ,text = "City")
cityLabel.grid(row = 12,column = 3)


pincodeLabel =  Label(window ,text = "Pincode")
pincodeLabel.grid(row = 14,column = 3)

#User Entries
#================ MENTIONING THE DATATYPES OF INPUT VA
RIABLES============

userNameEntry=StringVar()
vehicleNumberEntry = StringVar()
```

```
bloodGroupEntry=StringVar()
contactNumberEntry=StringVar()
emergencyNumberEntry=StringVar()
stateEntry=StringVar()
cityEntry=StringVar()
pincodeEntry=StringVar()

#=========== Defining the INPUT FIELDS=============


e1 = Entry(window,textvariable=userNameEntry)
e1.grid(row = 0,column = 6)

e2 = Entry(window,textvariable=vehicleNumberEntry)
e2.grid(row = 2,column = 6)

e3= Entry(window,textvariable=bloodGroupEntry)
e3.grid(row = 4,column = 6)

e4 = Entry(window,textvariable=contactNumberEntry)
e4.grid(row = 6,column = 6)

e5 = Entry(window,textvariable=emergencyNumberEntry)
e5.grid(row = 8,column = 6)

e6 = Entry(window,textvariable=stateEntry)
e6.grid(row = 10,column = 6)

e7= Entry(window,textvariable=cityEntry)
e7.grid(row = 12,column = 6)

e8= Entry(window,textvariable=pincodeEntry)
e8.grid(row = 14,column = 6)
```

```
def insert_data():

   # USING THE GET METHOD TO GET THE VALUES FROM THE I
NPUT FIELDS

    userName = userNameEntry.get()
    vehicleNumber=vehicleNumberEntry.get()
    bloodGroup=bloodGroupEntry.get()
    contactNumber=contactNumberEntry.get()
    emergencyNumber=emergencyNumberEntry.get()
    state=stateEntry.get()
    city=cityEntry.get()
    pincode=pincodeEntry.get()


    mydb=mysql.connector.connect(

        host='localhost',
        user='root',
        passwd='bmwm53552',
        database='pycode'

        )
    myCursor= mydb.cursor()

    myCursor.execute('INSERT INTO userregisteration (N
ame,VIN,Blood_Group,Contact_Number,Emergency_Number,St
ate,City,Pincode) VALUES(%s,%s,%s,%s,%s,%s,%s,%s)',
```

```
    (   userName,
        vehicleNumber,
        bloodGroup,
        contactNumber,
        emergencyNumber,
        state,
        city,
        pincode,



    )

    )

    mydb.commit()  #SAVING THE CHANGES IN THE TABLES
    mydb.close()    #CLOSING THE CONNECTION TO MYSQL


def crashMsg():



    userName = userNameEntry.get()
    vehicleNumber=vehicleNumberEntry.get()
    bloodGroup=bloodGroupEntry.get()
    contactNumber=contactNumberEntry.get()
    emergencyNumber=emergencyNumberEntry.get()
    state=stateEntry.get()
    city=cityEntry.get()
    pincode=pincodeEntry.get()
```

```
    emergencyNumber=emergencyNumberEntry.get()
    emergencyMessage=f'Your Relative {userName} with V
ehicle number {vehicleNumber} is \
                        in emergency at the following
Location :{latitude},{longitude} \
                            {pincode} , {place} ,{map}
 please send some Help!'
    roadSideMessage=f'A Vehicle {vehicleNumber} has so
meEmergency at following Loc \
                        {latitude},{longitude} \
                            {pincode} , {place},{map}'


    w  = send_sms(emergencyNumber,emergencyMessage)
    rsa= send_sms(8850652504,roadSideMessage)

    if w or rsa:
        showinfo('Success','Relax!!,Someone will reach
out to youu very soon :)')

    else:
        showerror('There Seems to be a Network Issue')





def nextScreen():
    # if userNameEntry.get() or vehicleNumberEntry.get
()  or contactNumberEntry.get() or emergencyNumberEntr
```

```
y.get() or stateEntry.get() or cityEntry.get() or pinc
odeEntry.get() == ' ':
    #     messagebox.showerror('Registeration Failed',
'All Fields are Mandatory')
    # else:

    userName = userNameEntry.get()
    contactNumber=contactNumberEntry.get()
    messagebox.showinfo('Success')

    window1 = Tk()
    window1.title("View Statics of car")
    window1.geometry('400x400')
    window1.configure(background = "#275296")

    #Label widgets
    batteryStatusLabel = Label(window1 ,text = "Logged
 in as")
    batteryStatusLabel.grid(row = 0,column = 0)

    speedVariationLabel = Label(window1 ,text = "Phone
 Number")
    speedVariationLabel.grid(row = 2,column = 0)

    loginName = Label(window1,text= userName)
    loginName.grid(row = 0,column = 1)

    phoneNumber = Label(window1,text = contactNumber)
    phoneNumber.grid(row = 2,column = 1)

    btn1 =ttk.Button(window1 ,text="Click For SpeedVar
iations",command=speedModule).grid(row=16,column=1)
```

```
    btn2 = ttk.Button(window1 ,text="Click For Battery
 Charging Status ",command=chargingGraph).grid(row=20,
column=1)
    btn3= ttk.Button(window1 ,text="Click For RoadSide
 24*7 Assistance ",command=crashMsg).grid(row=22,colum
n=1)
    btn4= ttk.Button(window1 ,text="Click For Live Spe
ed Graph ",command=speedPlot).grid(row=24,column=1)



    window1.mainloop()



btn = ttk.Button(window ,text="Submit",command=insert_
data).grid(row=22,column=6)
btnNext = ttk.Button(window ,text="Click to Proceed",c
ommand=nextScreen).grid(row=28,column=6)

#CREATING THE SUBMIT BUTTON

window.mainloop()

 #TERMINATING THE GUI LOOP



'''
VARIABLES USED :
```

```
1.    myCursor  : USED TO EXECUTE THE QUERIES INTO MYS
QL
2.    userName  : INPUT FROM THE USER
3.    vehicleNumber : VEHICLE NUMBER
4.    bloodGroup : BLOOD GROUP
5.    contactNumber : TO SEND THE MESSAGE
6.    emergencyNumber : TO SEND THE MESSAGE
7.    state : FOR REGISTERATION
8.    city : FOR REGISTERATION
9.    pincode : FOR REGISTERATION
10.   loginName  : To display on Screen 2
11.   phoneNumber : SAME AS ABOVE
12.   e1    : FIRST ENTRY SECTION IN GUI1
13.   e2    : SECOND ENTRY SECTION IN GUI1
14.   e3    : THIRD ENTRY SECTION IN GUI1
15.   e4    : FOURTH ENTRY SECTION IN GUI1
16.   e5    : FIFTH ENTRY SECTION   IN GUI1
17.   e6    : SIXTH ENTRY SECTION IN GUI1
18.   e7    : SEVENT ENTRY SECTION IN GUI1
19.   emergencyNumber : TO SEND MESSAGE TO THE PROVIDE
D NUMBER
20.   emergencyMessage: TO SEND THE EMERGENCY MESSAGE
21.   roadSideMessage : TO SEND THE MESSAGE TO 24X7 RO
ADSIDE ASSISTANCE
22.   w : TO CHECK STATUS OF THE MESSAGE SENT TO USER
PROVIDED NUMBER
23.   rsa : : TO CHECK STATUS OF THE MESSAGE SENT TO 2
4X7 ROADSIDE ASSISTANCE
24.   btn : SUBMIT BUTTON ON GUI 1
25.   btn1 : Click For SpeedVariations BUTTON
26.   btn2 : Click For Battery Charging Status
27.   btn3 : Click For RoadSide 24*7 Assistance
28.   btn4 : Click For Live Speed Graph
29.   btnNext : PROCEED TO NEXT GUI
```

```
30.   data : RECIEVE DATA IN FORM OF JAVASCRIPT OBJECT
 NOTATION(JSON) (WEBSCRAPPING)
31.   location : FOR LIVE LOCATION
32.   latitude : FOR LATITUDAL CO-ORDINATE
33.   longitude : FOR LONGITUDAL CO-ORDINATE
34.   place : FOR CITY DETAILS
35.   pincode : FOR POSTAL CODE DETAILS
36.   e8     : EIGHTH ENTRY SECTION IN GUI1
37.   charging : READ DATA FROM THE CSV FILE
38.   X : APPEND THE DATA FROM 1ST COLUMN
39.   Y:  APPEND THE DATA FROM 2ND COLUMN
40.   Z:  APPEND THE DATA FROM 3RD COLUMN

41.   deceleration=[]
42.   timee
'''
```

IV. Output

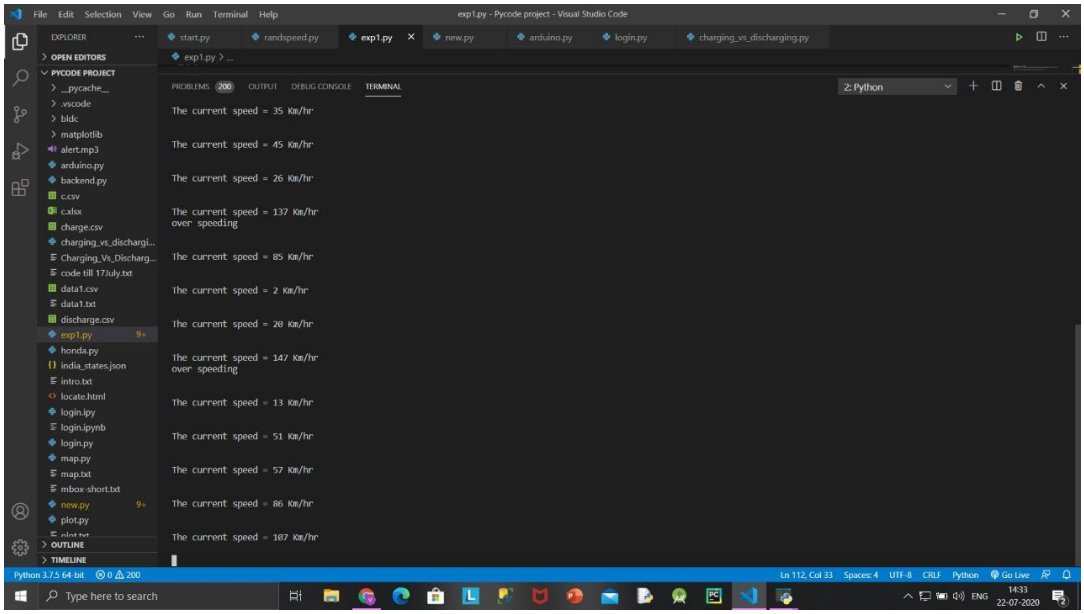

Fig.1

Displaying speed on the terminal

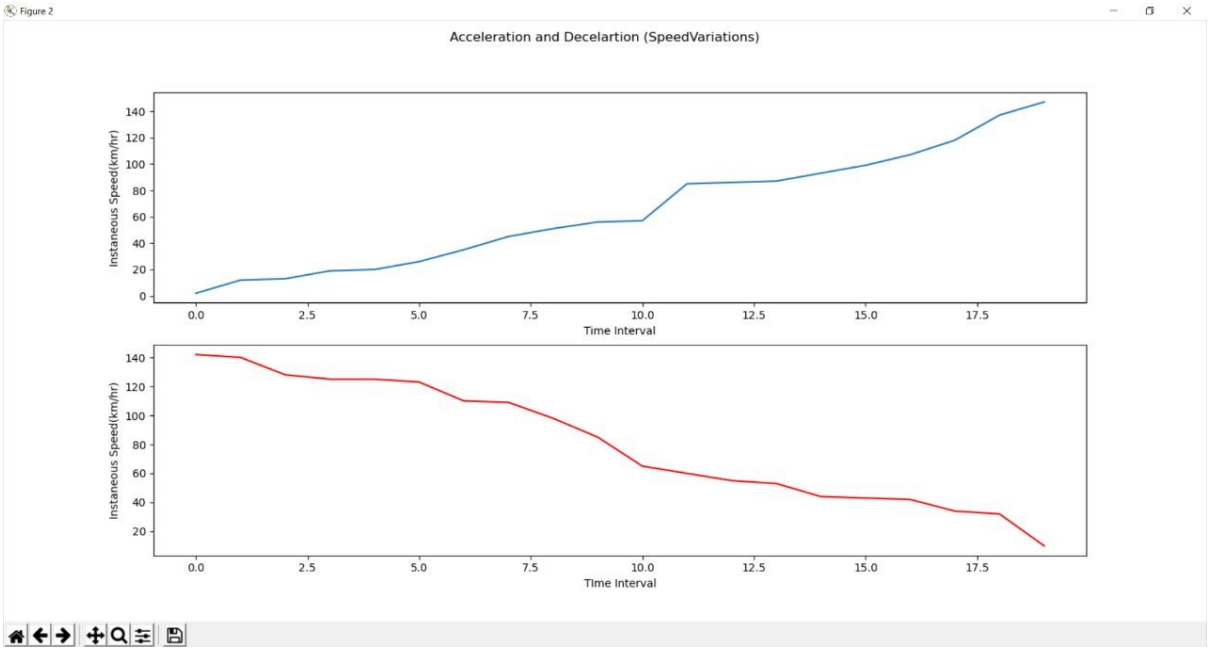

Fig.2

Displaying acceleration and deacceleration

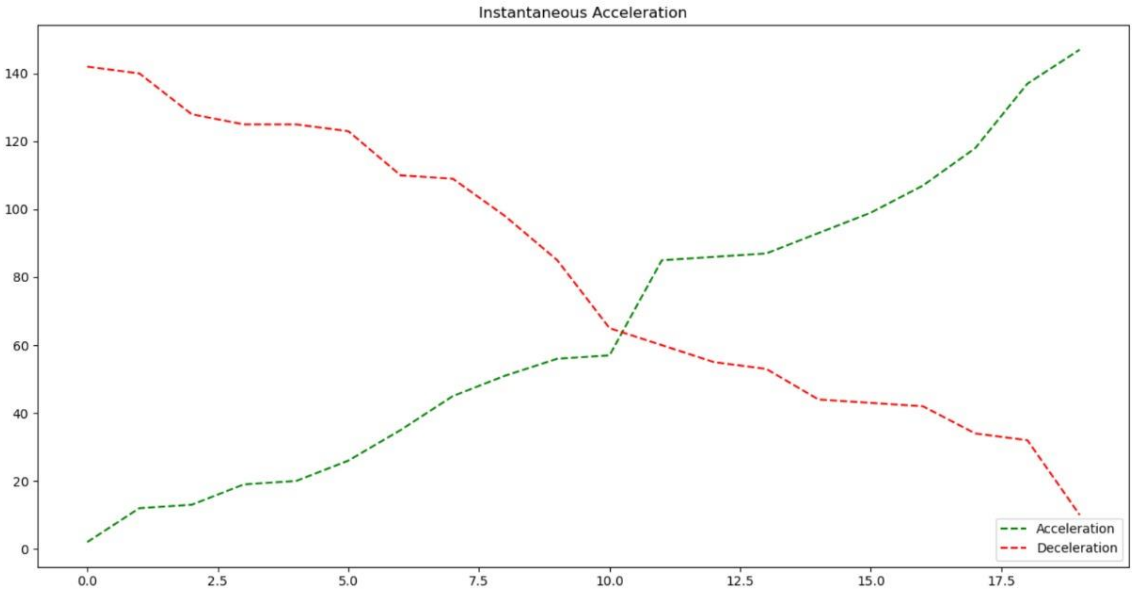

Fig.3

Displaying instantaneous acceleration

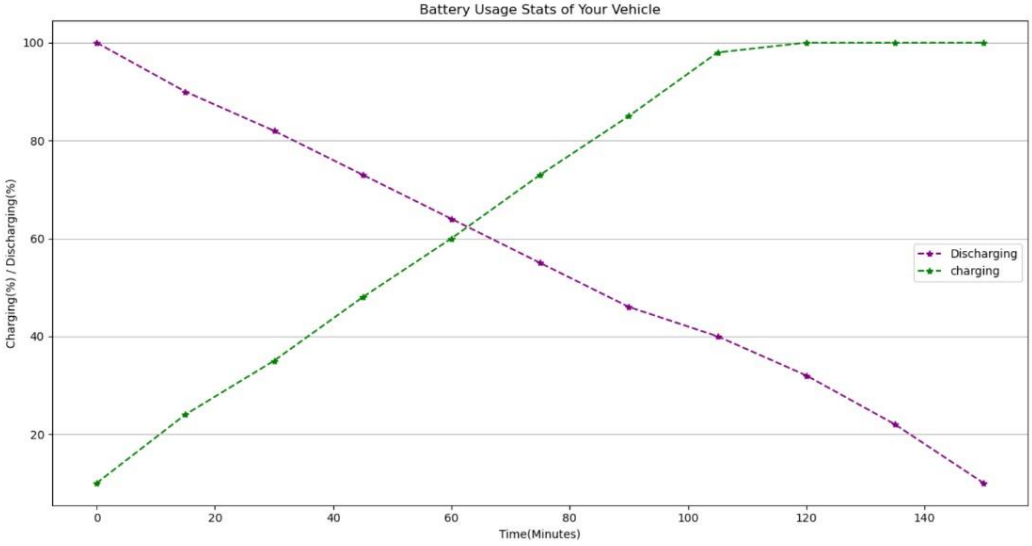

Fig.4

Displaying battery usage statistics of the vehicle using Matplotlib

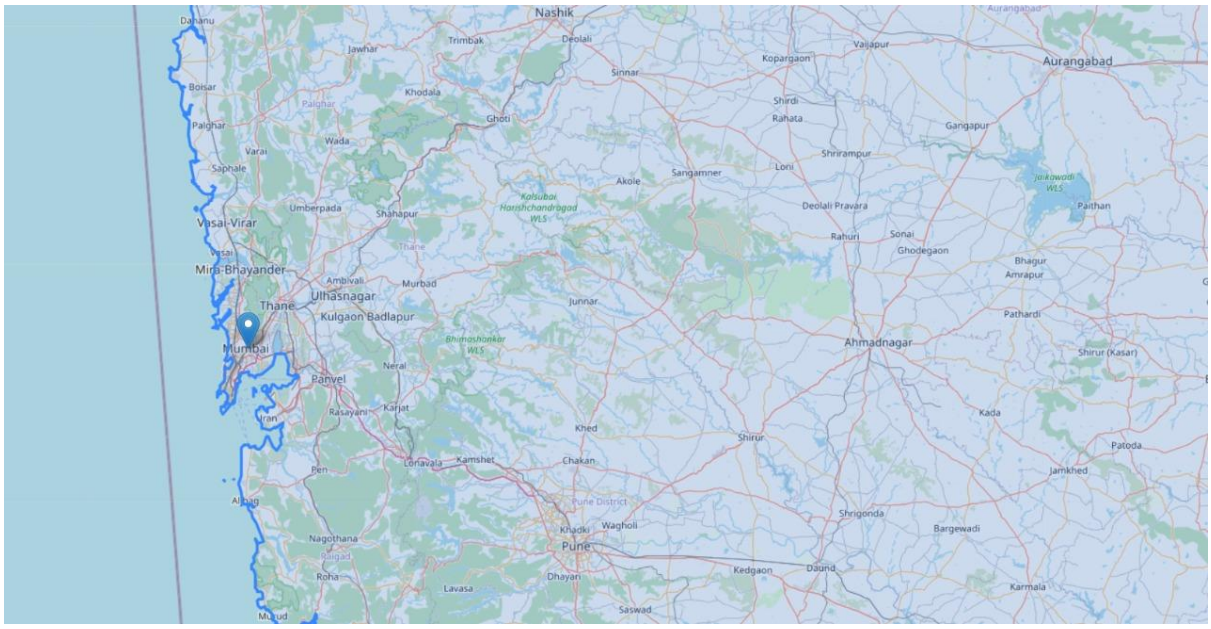

Fig.5

Locating on map

Locating the live location on the map (USING API)

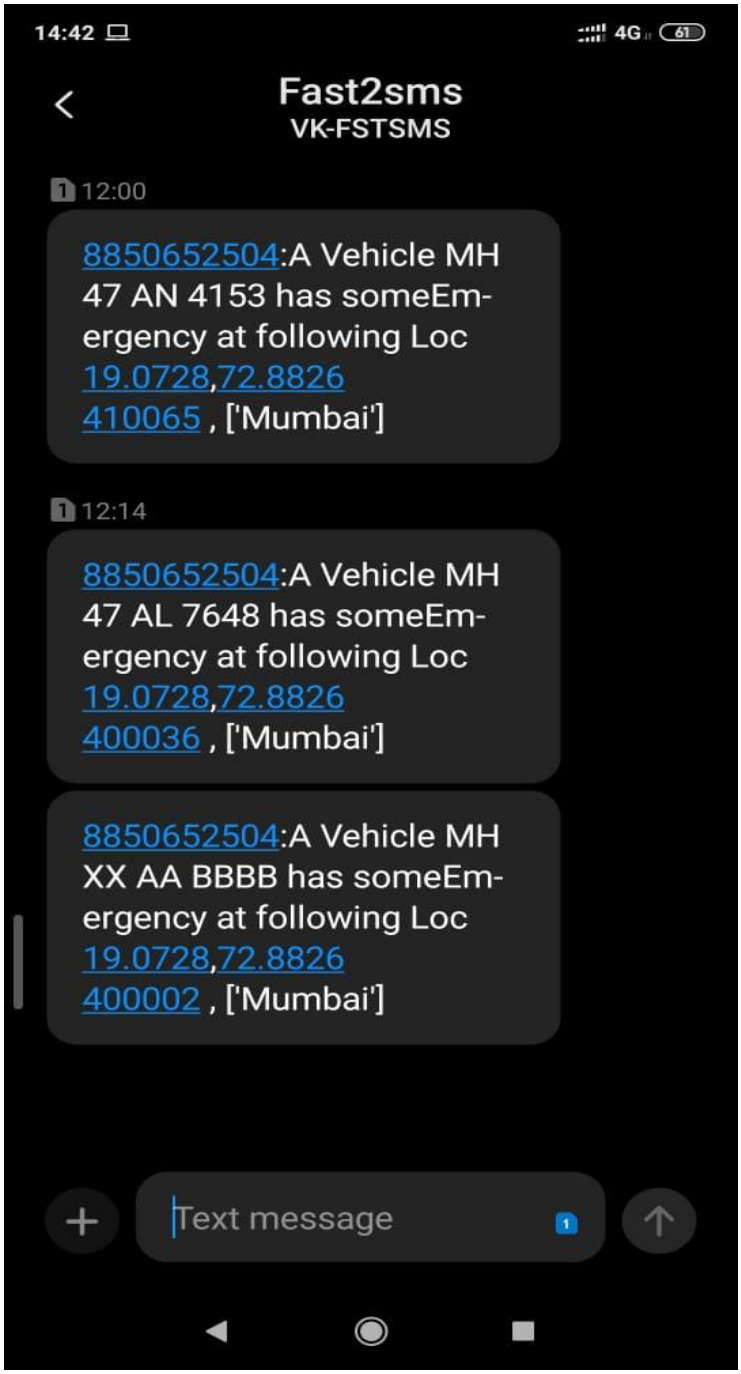

Fig.6

Message to roadside assistance

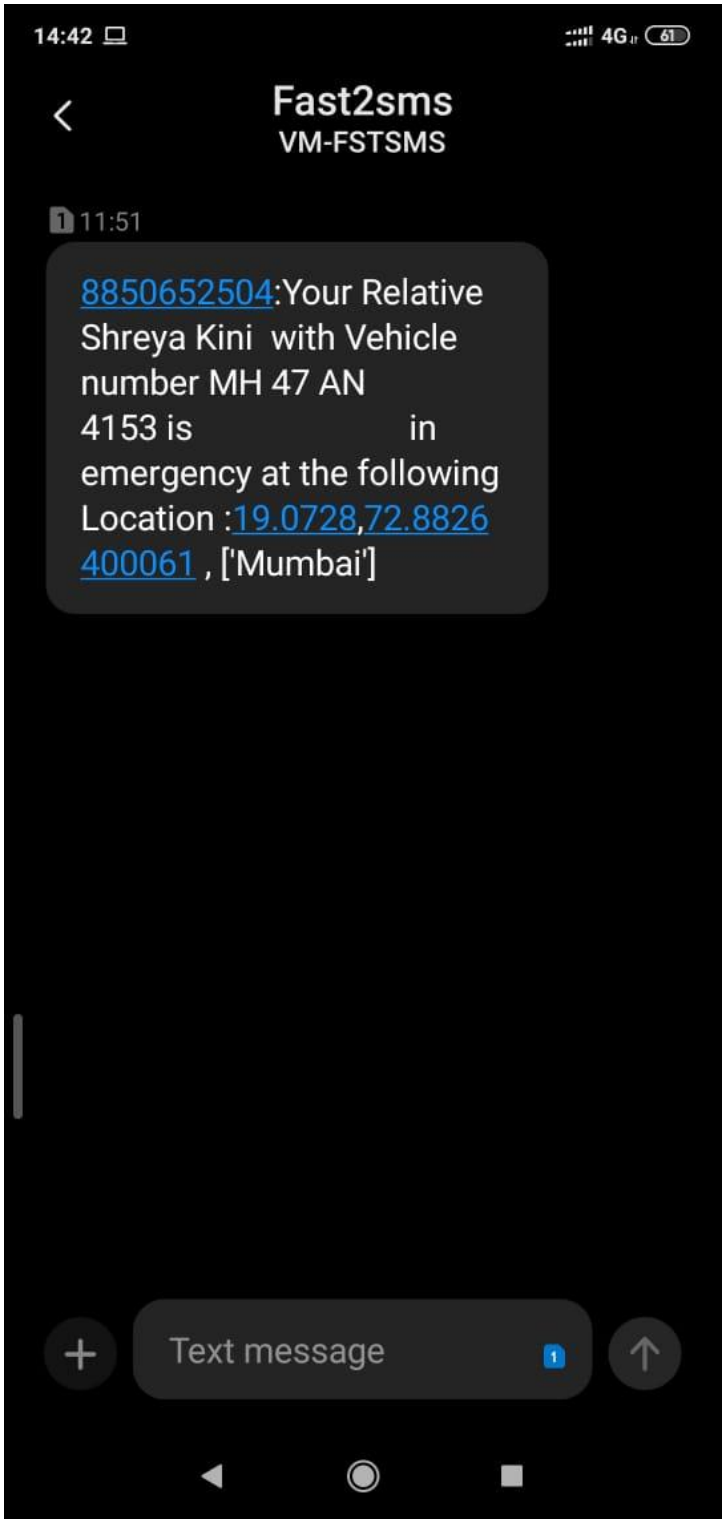

Fig.7

Message to relative

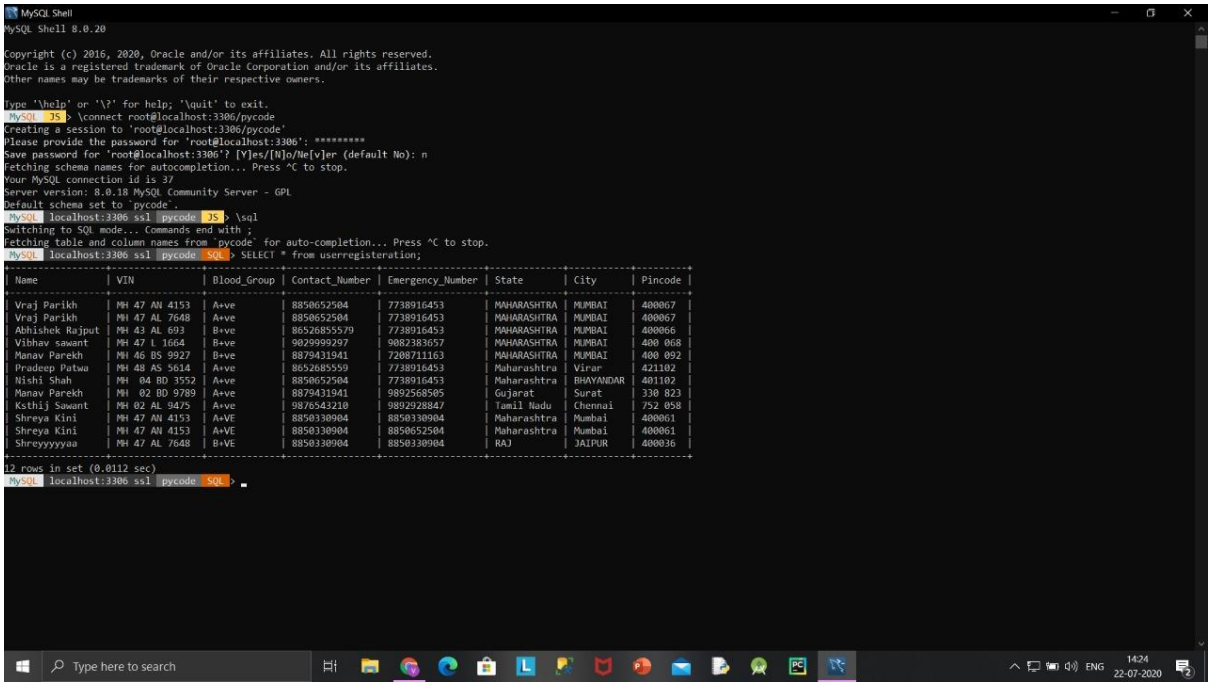

Fig.8

MySQL Database Shell

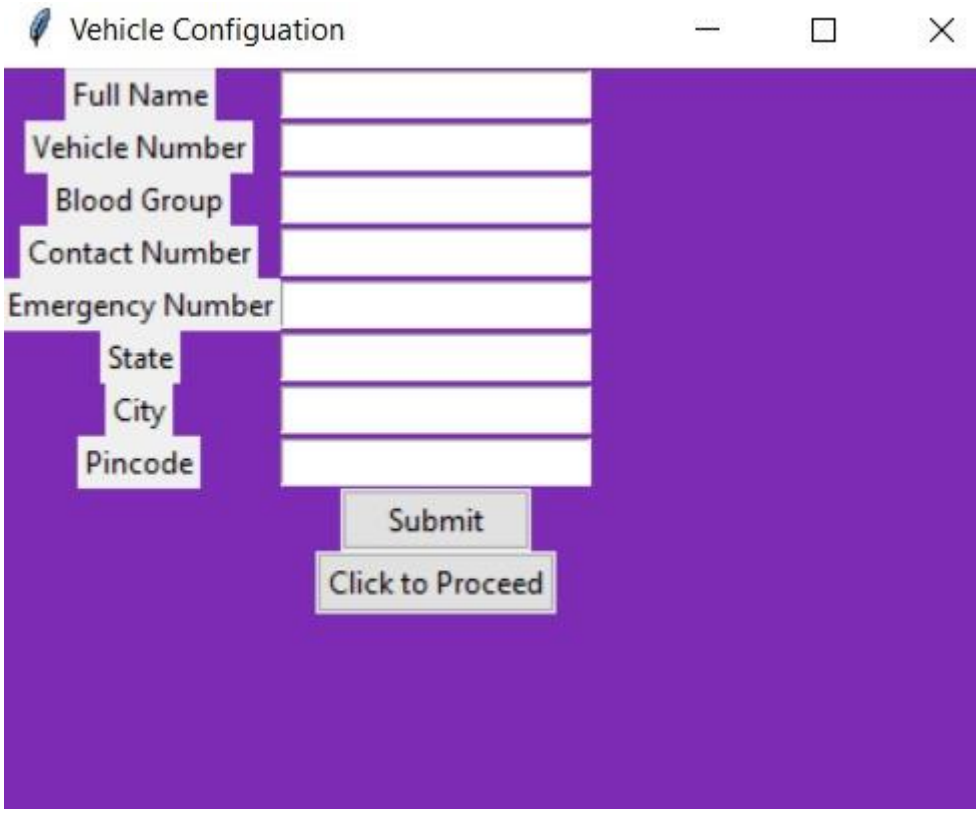

Fig.9

GUI 1: Vehicle Configuration

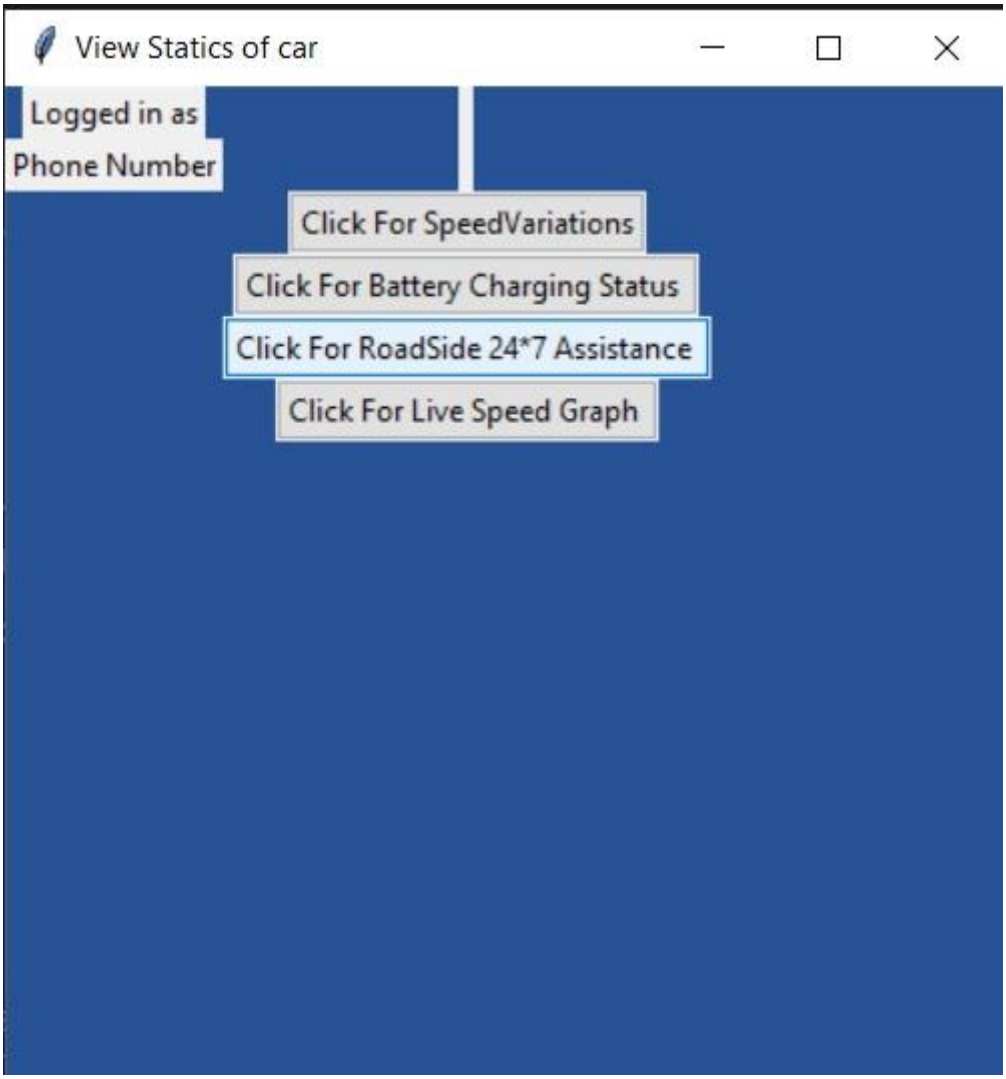

Fig.10

GUI 2 : Statics Navigation

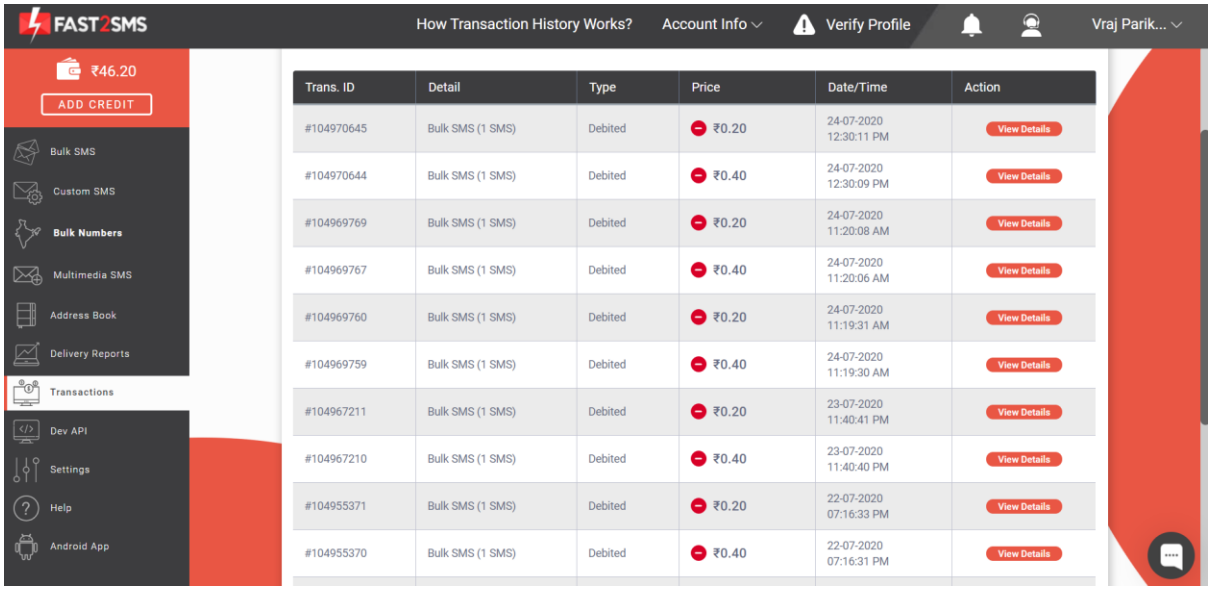

Fig: 11

Transcation details using FAST2SMS

Links And Software's:

MySQL Database	www.mysql.com
Arduino Ide	www.arduino.cc
FAST2SMS	***https://www.fast2sms.com/dashboard/sms/bulk***
Ip-info maps	*https://ipinfo.io/*
Visual Studio Code	*https://code.visualstudio.com/*
CMD	*For installing libraries*