PYTHON PROJECT REPORT (INT 213)

ON

PARKING MANAGEMENT SYSTEM

SUBMITTED BY

S.NO	ROLL NO	NAME OF THE STUDENT	REG NO
01	RK20RRBB64	JEEVAN JYOTHI KUMAR	**12018267**

JEEVAN JYOTHI KUMAR – PROGRAMMING

JEEVAN JYOTHI KUMAR –REPORT

PARKING MANAGEMENT SYSTEM

INTRODUCTION:

In this project we made a parking management system interface using graphical user interface, functions, files in python. In the the user gives the details of his/her name ,vehicle number, entry time, exit time, by this info we will calculate the parking fee and display the details of vehicle and display the parking fee to be paid. And all the details will be stored in files.

EXPLANATION:

GRAPHICAL USER INTERFACE:

Python offers multiple options for developing GUI (Graphical User Interface). Out of all the 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. Creating a GUI using tkinter is an easy task.

There are two main methods used which the user needs to remember while creating the Python application with GUI.

1. **Tk(screenName=None, baseName=None, className='Tk', useTk=1):** To create a main window, tkinter offers a method 'Tk(screenName=None, baseName=None, className='Tk', useTk=1)'. To change the name of the window, you can change the className to the desired one.
2. **mainloop():** There is a method known by the name mainloop() is used when your application is ready to run. mainloop() is an infinite loop used to run the application, wait for an event to occur and process the event as long as the window is not closed.

tkinter also offers access to the geometric configuration of the widgets which can organize the widgets in the parent windows. There are mainly three geometry manager classes class.

1. **pack() method:** It organizes the widgets in blocks before placing in the parent widget.
2. **grid() method:** It organizes the widgets in grid (table-like structure) before placing in the parent widget.
3. **place() method:** It organizes the widgets by placing them on specific positions directed by the programmer.

There are several widgets which you can put in your tkinter application,

- **Button**: To add a button in your application, this widget is used.
- **Canvas:** It is used to draw pictures and other complex layout like graphics, text and widgets.
- **Check Button:** To select any number of options by displaying a number of options to a user as toggle buttons.
- **Entry:** It is used to input the single line text entry from the user.. For multi-line text input, Text widget is used.
- **Label**: It refers to the display box where you can put any text or image which can be updated any time as per the code.

- **List box**: It offers a list to the user from which the user can accept any number of options.

FUNCTIONS IN PYTHON :

A function is a block of organized, reusable code that is used to perform a single, related action. Functions provide better modularity for your application and a high degree of code reusing.

As we already know, Python gives you many built-in functions like print(), etc. but you can also create your own functions. These functions are called *user-defined functions.*

Defining a Function

You can define functions to provide the required functionality. Here are simple rules to define a function in Python.

- Function blocks begin with the keyword **def** followed by the function name and parentheses (( )).
- Any input parameters or arguments should be placed within these parentheses. You can also define parameters inside these parentheses.
- The first statement of a function can be an optional statement - the documentation string of the function or *docstring*.
- The code block within every function starts with a colon (:) and is indented.
- The statement return [expression] exits a function, optionally passing back an expression to the caller. A return statement with no arguments is the same as return None.

Calling a Function

Defining a function only gives it a name, specifies the parameters that are to be included in the function and structures the blocks of code.

Once the basic structure of a function is finalized, you can execute it by calling it from another function or directly from the Python prompt. Following is the example to call printme() function

Function Arguments

We can call a function by using the following types of formal arguments ,

- Required arguments
- Keyword arguments
- Default arguments
- Variable-length arguments

FILES IN PYTHON :

Python provides basic functions and methods necessary to manipulate files by default. You can do most of the file manipulation using a **file** object.

The open Function

Before you can read or write a file, you have to open it using Python's built-in *open()* function. This function creates a **file** object, which would be utilized to call other support methods associated with it.

Some are parameter details −

- **file_name** − The file_name argument is a string value that contains the name of the file that you want to access.
- **access_mode** − The access_mode determines the mode in which the file has to be opened, i.e., read, write, append, etc. A complete list of possible values is given below in the table. This is optional parameter and the default file access mode is read (r).
- **buffering** − If the buffering value is set to 0, no buffering takes place. If the buffering value is 1, line buffering is performed while accessing a file. If you specify the buffering value as an integer greater than 1, then buffering action is performed with the indicated buffer size. If negative, the buffer size is the system default(default behaviour).

Different modes of opening a file

1. r

Opens a file for reading only. The file pointer is placed at the beginning of the file. This is the default mode.

2. w

Opens a file for writing only. Overwrites the file if the file exists. If the file does not exist, creates a new file for writing.

3. a

Opens a file for appending. The file pointer is at the end of the file if the file exists. That is, the file is in the append mode. If the file does not exist, it creates a new file for writing.

4. ab

Opens a file for appending in binary format. The file pointer is at the end of the file if the file exists. That is, the file is in the append mode. If the file does not exist, it creates a new file for writing.

The file Object Attributes

Once a file is opened and you have one *file* object, you can get various information related to that file.

Attributes related to file object

s.no	Command	description
01	**file.closed**	Returns true if file is closed, false otherwise.
02	**file.mode**	Returns access mode with which file was opened.
03	**file.name**	Returns name of the file.
04	**file.softspace**	Returns false if space explicitly required with print, true otherwise.

The close () Method

The close() method of a *file* object flushes any unwritten information and closes the file object, after which no more writing can be done.

Python automatically closes a file when the reference object of a file is reassigned to another file. It is a good practice to use the close () method to close a file.

Reading and Writing Files

The *file* object provides a set of access methods to make our lives easier. We would see how to use *read()* and *write()* methods to read and write files.

Opening a File

It is done using the `open ()` function. No module is required to be imported for this function.

note

The file should exist in the same directory as the python program file else, full address of the file should be written on place of filename.

SOURCE CODE:

```
import tkinter
from tkinter import *
from tkinter import ttk
import time

root = tkinter.Tk()  #defining main window
root.title("Parking Management System")
root.geometry('1028x400')
root.config(bg = "#091833")
root.resizable(0,0)
#fonts
font11 = "{U.S.101} 20 bold"
font2 = "{Segoe UI} 13 bold"
font3 = "Arial 13 bold"
font4 = "{Courier New} 10 bold"
font5 = "{Courier New} 10 normal"

localtime = time.asctime(time.localtime(time.time()))
#defining labels

a1 = tkinter.Label(root,text="PARKING MANAGEMENT SYSYTEM",
bg="#091833",fg="orange",font=font11)
a1.place(relx = 0.268, rely = 0.02, height = 51, width=507)
a2 = tkinter.Label(root,text="Select the type of vehicle",bg="#091833",fg="white",font= font2)
a2.place(x = 410, y = 90)
a = tkinter.Label(root,text=localtime,bg="#091833",fg="steel blue",font=font2)
a.place(x = 410, y = 50)

#define the fuction
def fourwheels():

    #defining another window
    newwindow = tkinter.Toplevel(root)
    newwindow.title("Four wheeler Parking")
    newwindow.geometry('500x400+600+40')
    newwindow.config(bg="#091833")

    a3 = tkinter.Label(newwindow,text="Four wheeler parking details",bg="#091833",fg="white",
font=("Aries",15,"bold"))
    a3.place(x = 200, y = 25)
    a4 = tkinter.Label(newwindow,text = "Enter the vehicle number",bg = "#091833",fg = "white", font =
font2)
    a4.place(x =150 , y = 60)

    num_entry = StringVar()
    num_entry = tkinter.Entry(newwindow,textvariable = num_entry,width = 25)
    num_entry.place(x = 400, y = 65)

    a5 = tkinter.Label(newwindow,text = "Enter the name of the driver",bg = "#091833",fg = "white",
font = font3)
    a5.place(x =150 , y = 95)

    name_entry = StringVar()
    name_entry = tkinter.Entry(newwindow,textvariable = name_entry,width = 25)
    name_entry.place(x = 400, y = 100)

    a6 = tkinter.Label(newwindow,text="Time at which vehicle entered ",bg = "#091833",fg = "white",
font = font3)
    a6.place(x =150 , y = 130)

    time_entry1 = IntVar()
    time_entry1 = tkinter.Entry(newwindow,textvariable = time_entry1,width = 25)
    time_entry1.place(x = 400, y = 135)

    a7 = tkinter.Label(newwindow,text = "Time at which vehicle left ",bg = "#091833",fg = "white", font
= font3)
    a7.place(x =150 , y = 165)

    time_entry2 = IntVar()
    time_entry2 = tkinter.Entry(newwindow,textvariable = time_entry2,width = 25)
    time_entry2.place(x = 400, y = 170)

    def print_bill():
        num = float(time_entry1.get())
        num0 = float(time_entry2.get())
        result = (num0-num)*50
        total = str(result)

        p = tkinter.Label(newwindow,text="vehicle number: " + num_entry.get(),bg = "#091833",fg =
"white",font = font3, width = 30)
        p.place(x = 250, y = 230)

        p1 = tkinter.Label(newwindow,text="vehicle owner: " + name_entry.get(),bg = "#091833",fg =
"white",font = font3,width = 30)
        p1.place(x = 250, y = 255)

        p2 = tkinter.Label(newwindow,text="vehicle entry time: " + time_entry1.get(),bg = "#091833",fg =
"white",font = font3, width = 30)
        p2.place(x = 250, y = 280)
        p3 = tkinter.Label(newwindow,text="vehicle exit time: " + time_entry2.get(),bg = "#091833",fg = 
"white",font = font3,width = 30)
        p3.place(x = 250, y = 310)

        p4 = tkinter.Label(newwindow,text = "vehicle parking fee: " + total,bg = "#091833",fg = 
"white",font = font3,width = 30)
        p4.place(x = 250, y = 340)

        def save_info():

            reg_no = num_entry.get()
            driver_name = name_entry.get()
            entry_time = time_entry1.get()
            exit_time = time_entry2.get()
            parking_fee = total

            print("Vehicle number",reg_no)
            print("Vehicle driver name",driver_name)
            print("Vehicle entry time",entry_time)
            print("Vehicle exit time",exit_time)

            file = open("four.txt","a")

            file.write("vehicle registration number: " + reg_no)

            file.write("\n")

            file.write("vehicle driver name: " + driver_name)

            file.write("\n")

            file.write("vehicle entering time: " + str(entry_time))

            file.write("\n")

            file.write("vehicle leaving time: " + str(exit_time))

            file.write("\n")

            file.write("Vehicle parking fee: " + parking_fee)

            file.write("\n")

            file.close()
        return save_info()

    bill_btn1 = tkinter.Button(newwindow,text="Print bill",command=print_bill)
    bill_btn1.place(x = 400, y = 200)

def twowheels():

    #defining the another window
    newwindow1 = tkinter.Toplevel(root)
    newwindow1.title("Two wheeler Parking")
    newwindow1.geometry('500x300+400+60')
    newwindow1.config(bg="#091833")

    b = tkinter.Label(newwindow1,text="Details of Two wheeler
Parking",bg="#091833",fg="white",font=font2)
    b.place(x = 200, y = 25)

    b1 = tkinter.Label(newwindow1,text="Enter the vehicle
number",bg="#091833",fg="white",font=font2)
    b1.place(x =150 , y = 60)

    text_entry1 = tkinter.Entry(newwindow1,width=25)
    text_entry1.place(x = 400, y = 65)

    b2 = tkinter.Label(newwindow1,text="Enter the name of the
driver",bg="#091833",fg="white",font=font2)
    b2.place(x =150 , y = 95)

    text_entry2 = tkinter.Entry(newwindow1,width=25)
    text_entry2.place(x = 400, y = 100)

    b3 = tkinter.Label(newwindow1,text="Time at which vehicle entered
",bg="#091833",fg="white",font=font2)
    b3.place(x =150 , y = 130)

    time_entry3 = tkinter.Entry(newwindow1,width=25)
    time_entry3.place(x = 400, y = 135)

    b4 = tkinter.Label(newwindow1,text="Time at which vehicle left
",bg="#091833",fg="white",font=font2)
    b4.place(x =150 , y = 165)

    time_entry4 = tkinter.Entry(newwindow1,width=25)
    time_entry4.place(x = 400, y = 170)
```

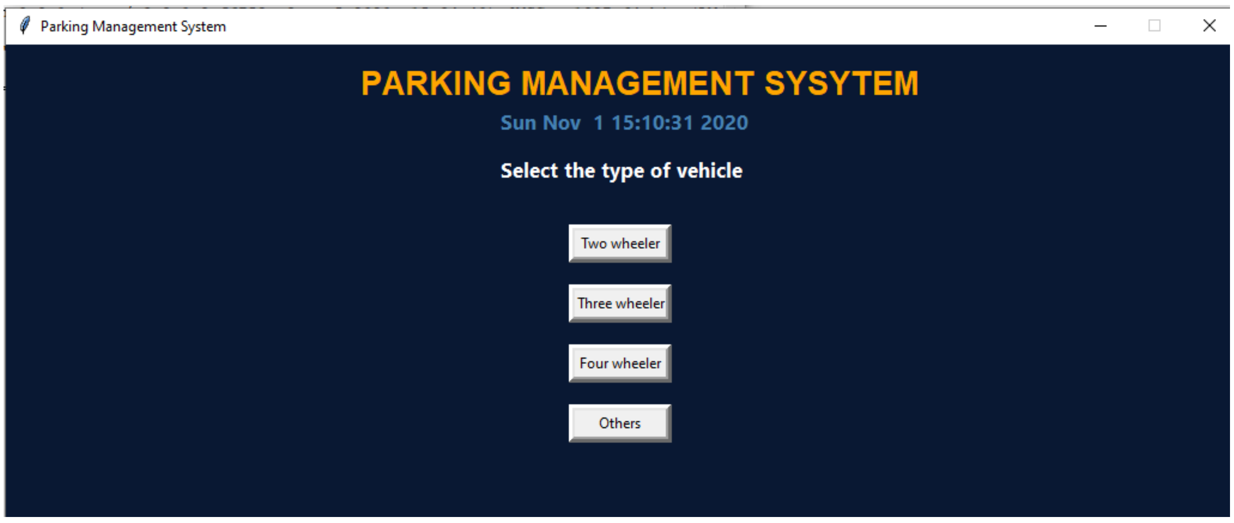
```
  def print_bill1():
      num1 = float(time_entry3.get())
      num2 = float(time_entry4.get())
      result = (num2-num1)*25
      total = str(result)

      q = tkinter.Label(newwindow1,text="vehicle number: " +
text_entry1.get(),bg="#091833",fg="white",font=("Aries",10), width= 30)
      q.place(x = 250, y = 230)

      q1 = tkinter.Label(newwindow1,text="vehicle owner: " +
text_entry2.get(),bg="#091833",fg="white",font=("Aries",10),width = 30)
      q1.place(x = 250, y = 255)

      q2 = tkinter.Label(newwindow1,text="vehicle entry time: " +
time_entry3.get(),bg="#091833",fg="white",font=("Aries",10), width = 30)
      q2.place(x = 250, y = 280)

      q3 = tkinter.Label(newwindow1,text="vehicle exit time: " +
time_entry4.get(),bg="#091833",fg="white",font=("Aries",10),width = 30)
      q3.place(x = 250, y = 310)

      q4 = tkinter.Label(newwindow1,text="vehicle parking fee: " +
total,bg="#091833",fg="white",font=("Aries",10),width = 30)
      q4.place(x = 250, y = 340)

      def save_info2():
          reg_no = text_entry1.get()
          driver_name = text_entry2.get()
          entry_time = time_entry3.get()
          exit_time = time_entry4.get()
          parking_fee = total

          print("vehicle number:",reg_no)
          print("Vehicle driver name:",driver_name)
          print("Vehicle entry time:",entry_time)
          print("vehicle exit time",exit_time)
          print("parking fee:" , parking_fee)

          file = open("two.txt","a")

          file.write("vehicle registration number: " + reg_no)

          file.write("\n")
```

```
            file.write("vehicle driver name: " + driver_name)

            file.write("\n")

            file.write("vehicle entering time: " + str(entry_time))

            file.write("\n")

            file.write("vehicle leaving time: " + str(exit_time))

            file.write("\n")

            file.write("Vehicle parking fee: " + parking_fee)

            file.write("\n")

            file.close()

        return save_info2()

    bill_btn2 = tkinter.Button(newwindow1,text="Print bill",command=print_bill1)
    bill_btn2.place(x = 400, y = 200)

def threewheels():

    #defining another window
    newwindow2 = tkinter.Toplevel(root)
    newwindow2.title("Three wheeler Parking")
    newwindow2.geometry('500x500+400+40')
    newwindow2.config(bg="#091833")

    c0 = tkinter.Label(newwindow2,text="Details of Three wheeler
Parking",bg="#091833",fg="white",font=font3)
    c0.place(x = 200, y = 25)

    c1 = tkinter.Label(newwindow2,text="Enter the vehicle
number",bg="#091833",fg="white",font=font2)
    c1.place(x =150 , y = 60)

    text_entry1 = StringVar()
    text_entry1 = tkinter.Entry(newwindow2,textvariable=text_entry1,width=25)
    text_entry1.place(x = 400, y = 65)

    c2 = tkinter.Label(newwindow2,text="Enter the name of the
driver",bg="#091833",fg="white",font=font2)
```

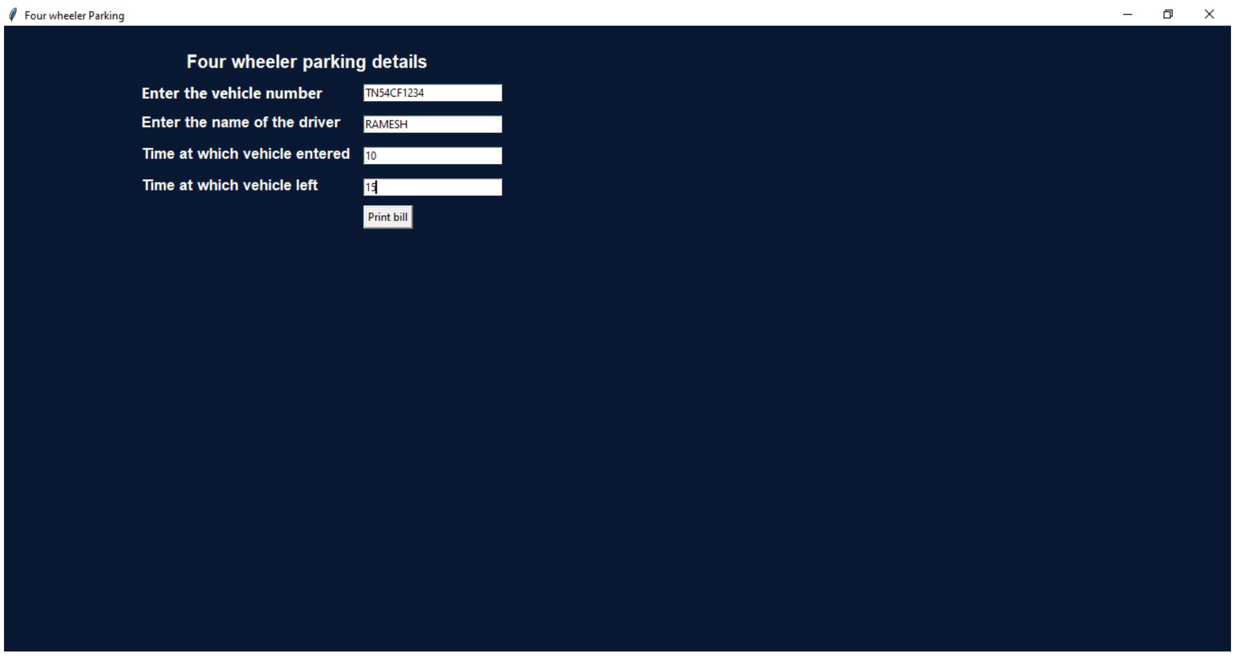
```
    c2.place(x =150 , y = 95)

    text_entry2 = StringVar()
    text_entry2 = tkinter.Entry(newwindow2,textvariable=text_entry2,width=25)
    text_entry2.place(x = 400, y = 100)

    c3 = tkinter.Label(newwindow2,text="Time at which vehicle
entered",bg="#091833",fg="white",font=font2)
    c3.place(x =150 , y = 130)

    time_entry3 = IntVar()
    time_entry3 = tkinter.Entry(newwindow2,textvariable=time_entry3,width=25)
    time_entry3.place(x = 400, y = 135)

    c4 = tkinter.Label(newwindow2,text="Time at which vehicle
left",bg="#091833",fg="white",font=font2)
    c4.place(x =150 , y = 165)

    time_entry4 = IntVar()
    time_entry4 = tkinter.Entry(newwindow2,textvariable=time_entry4,width=25)
    time_entry4.place(x = 400, y = 170)

    def print_bill2():
        num3 = float(time_entry3.get())
        num4 = float(time_entry4.get())
        result = (num4-num3)*35
        total = str(result)

        r = tkinter.Label(newwindow2,text="vehicle number: " +
text_entry1.get(),bg="#091833",fg="white",font=font4, width= 30)
        r.place(x = 250, y = 230)

        r1 = tkinter.Label(newwindow2,text="vehicle owner: " +
text_entry2.get(),bg="#091833",fg="white",font=font4,width = 30)
        r1.place(x = 250, y = 255)

        r2 = tkinter.Label(newwindow2,text="vehicle entry time: " +
time_entry3.get(),bg="#091833",fg="white",font=font4, width = 30)
        r2.place(x = 250, y = 280)

        r3 = tkinter.Label(newwindow2,text="vehicle exit time: " +
time_entry4.get(),bg="#091833",fg="white",font=font4,width = 30)
        r3.place(x = 250, y = 310)
        r4 = tkinter.Label(newwindow2,text="vehicle parking fee: " +
total,bg="#091833",fg="white",font=font4,width = 30)
        r4.place(x = 250, y = 340)

        def save_info3():

            reg_no = text_entry1.get()
            driver_name = text_entry2.get()
            entry_time = time_entry3.get()
            exit_time = time_entry4.get()
            parking_fee = total

            print("vehicle number:",reg_no)
            print("Vehicle driver name:",driver_name)
            print("Vehicle entry time:",entry_time)
            print("vehicle exit time",exit_time)
            print("parking fee:" , parking_fee)

            file = open("three.txt","a")

            file.write("vehicle registration number: " + reg_no)

            file.write("\n")

            file.write("vehicle driver name: " + driver_name)

            file.write("\n")

            file.write("vehicle entering time: " + str(entry_time))

            file.write("\n")

            file.write("vehicle leaving time: " + str(exit_time))

            file.write("\n")

            file.write("vehicle parking fee: " + parking_fee)

            file.write("\n")

            file.close()

        return save_info3()

    bill_btn3 = tkinter.Button(newwindow2,text="Print bill",command=print_bill2)
    bill_btn3.place(x = 400, y = 200)

def others():
    #defining another window
    newwindow3 = tkinter.Toplevel(root)
    newwindow3.title("Other Parking")
    newwindow3.geometry('750x450+400+40')
    newwindow3.config(bg="#091833")

    d0 = tkinter.Label(newwindow3,text="Details of other types of vehicle
Parking",bg="#091833",fg="white",font=font2)
    d0.place(x = 200, y = 25)

    d1 = tkinter.Label(newwindow3,text="Enter the vehicle
number",bg="#091833",fg="white",font=font2)
    d1.place(x =150 , y = 60)

    text_entry1 = StringVar()
    text_entry1 = tkinter.Entry(newwindow3,textvariable=text_entry1,width=25)
    text_entry1.place(x = 400, y = 65 )

    d2 = tkinter.Label(newwindow3,text="Enter the name of the
driver",bg="#091833",fg="white",font=font2)
    d2.place(x = 150, y = 95)

    text_entry2 = StringVar()
    text_entry2 = tkinter.Entry(newwindow3,textvariable=text_entry2,width=25)
    text_entry2.place(x = 400, y = 100)

    d3 = tkinter.Label(newwindow3,text = "Time at which vehicle entered ",bg = "#091833",fg =
"white",font = font2)
    d3.place(x = 150, y = 130)

    time_entry3 = IntVar()
    time_entry3 = tkinter.Entry(newwindow3,textvariable = time_entry3,width=25)
    time_entry3.place(x = 400, y = 135)

    d4 = tkinter.Label(newwindow3,text = "Time at which vehicle left ",bg = "#091833",fg = "white",font
= font2)
    d4.place(x = 150, y = 165)

    time_entry4 = IntVar()
    time_entry4 = tkinter.Entry(newwindow3,textvariable = time_entry4,width = 25)
    time_entry4.place(x = 400, y = 170)
    def print_bill():
        num1=float(time_entry3.get())
        num2=float(time_entry4.get())
        result=(num2-num1)*50
        total = str(result)

        s = tkinter.Label(newwindow3,text="vehicle number: " +
text_entry1.get(),bg="#091833",fg="white",font=font4, width= 30)
        s.place(x = 250, y = 230)

        s1 = tkinter.Label(newwindow3,text="vehicle owner: " +
text_entry2.get(),bg="#091833",fg="white",font=font4,width = 30)
        s1.place(x = 250, y = 255)

        s2 = tkinter.Label(newwindow3,text="vehicle entry time: " +
time_entry3.get(),bg="#091833",fg="white",font=font4, width = 30)
        s2.place(x = 250, y = 280)

        s3 = tkinter.Label(newwindow3,text="vehicle exit time: " +
time_entry4.get(),bg="#091833",fg="white",font=font4,width = 30)
        s3.place(x = 250, y = 310)

        s4 = tkinter.Label(newwindow3,text="vehicle parking fee: " +
total,bg="#091833",fg="white",font=font4,width = 30)
        s4.place(x = 250, y = 340)

        def save_info1():
            reg_no = text_entry1.get()
            driver_name = text_entry2.get()
            entry_time = time_entry3.get()
            exit_time = time_entry4.get()
            parking_fee = total

            print("vehicle number:",reg_no)
            print("Vehicle driver name:",driver_name)
            print("Vehicle entry time:",entry_time)
            print("vehicle exit time",exit_time)
            print("parking fee:" , parking_fee)

            file = open("others.txt","a")

            file.write("vehicle registration number: " + reg_no)
            file.write("\n")

            file.write("vehicle driver name: " + driver_name)
            file.write("\n")

            file.write("vehicle entering time: " + str(entry_time))

            file.write("\n")

            file.write("vehicle leaving time: " + str(exit_time))

            file.write("\n")

            file.write("parking fee: " + total)

            file.write("\n")

            file.close()

        return save_info1()

    bill_btn1 = tkinter.Button(newwindow3,text="Print bill",command=print_bill)
    bill_btn1.place(x = 400, y = 200)

#defining buttons

btn1 = tkinter.Button(root, text= "Two wheeler",borderwidth = 5,width = 10, command = twowheels)
btn1.place(x = 470, y = 150)

btn2 = tkinter.Button(root,text = "Three wheeler",borderwidth = 5,width = 10, command =
threewheels)
btn2.place(x = 470, y = 200)

btn3 = tkinter.Button(root,text = "Four wheeler",borderwidth = 5,width = 10,command = fourwheels)
btn3.place(x = 470, y = 250)

btn4 = tkinter.Button(root,text = "Others",borderwidth = 5,width = 10,command = others)
btn4.place(x = 470, y = 300)

root.mainloop()
```

GUI SCREENSHOTS:

1. WELCOME SCREEN :

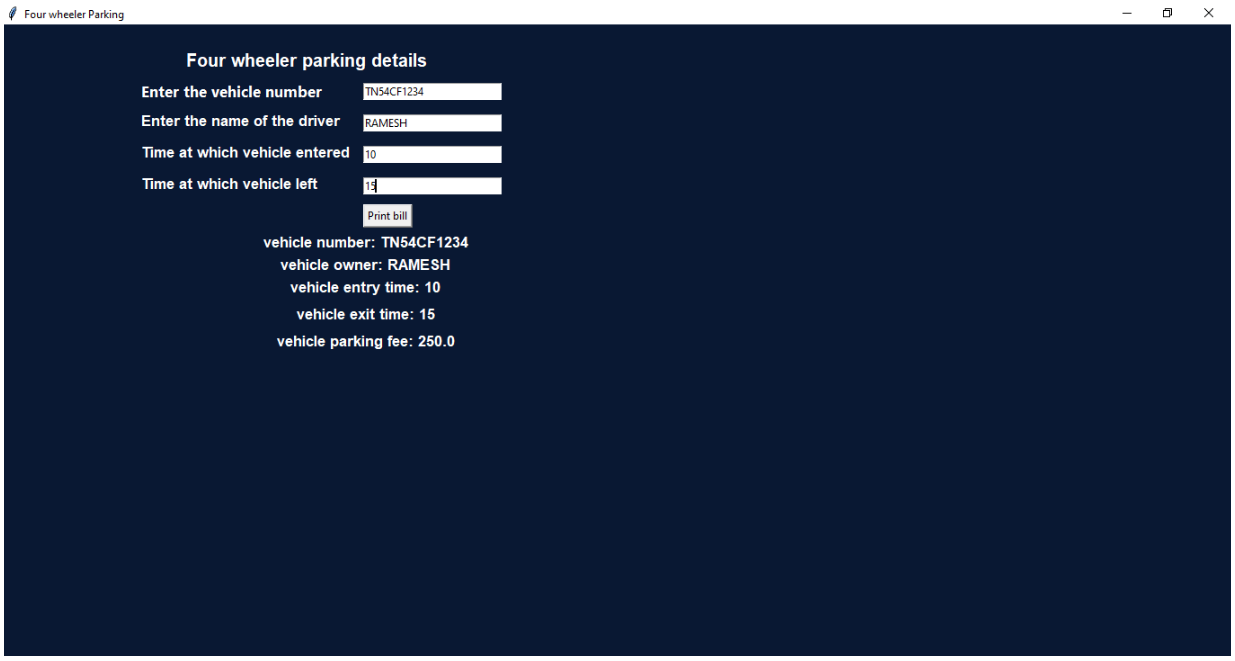

Parking Management System
PARKING MANAGEMENT SYSYTEM
Sun Nov 1 15:10:31 2020
Select the type of vehicle
Two wheeler
Three wheeler
Four wheeler
Others

2.FOUR-WHEELER (on clicking four-wheeler button)



Enter the vehicle number TN54CF1234
Enter the name of the driver RAMESH
Time at which vehicle entered 10
Time at which vehicle left 15

3.PRINTBILL(on clicking it display’s generated parking fee)



Four wheeler parking details

Enter the vehicle number TN54CF1234

Enter the name of the driver RAMESH

Time at which vehicle entered 10

Time at which vehicle left 15

vehicle number: TN54CF1234

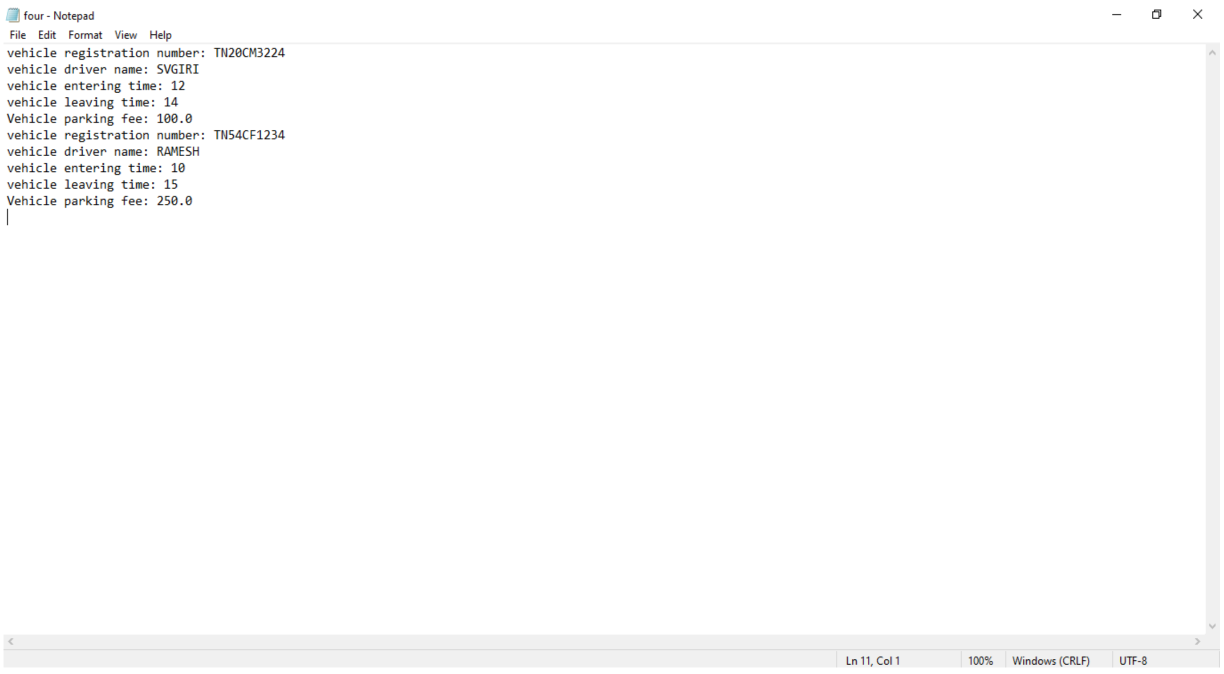
vehicle owner: RAMESH

vehicle entry time: 10

vehicle exit time: 15

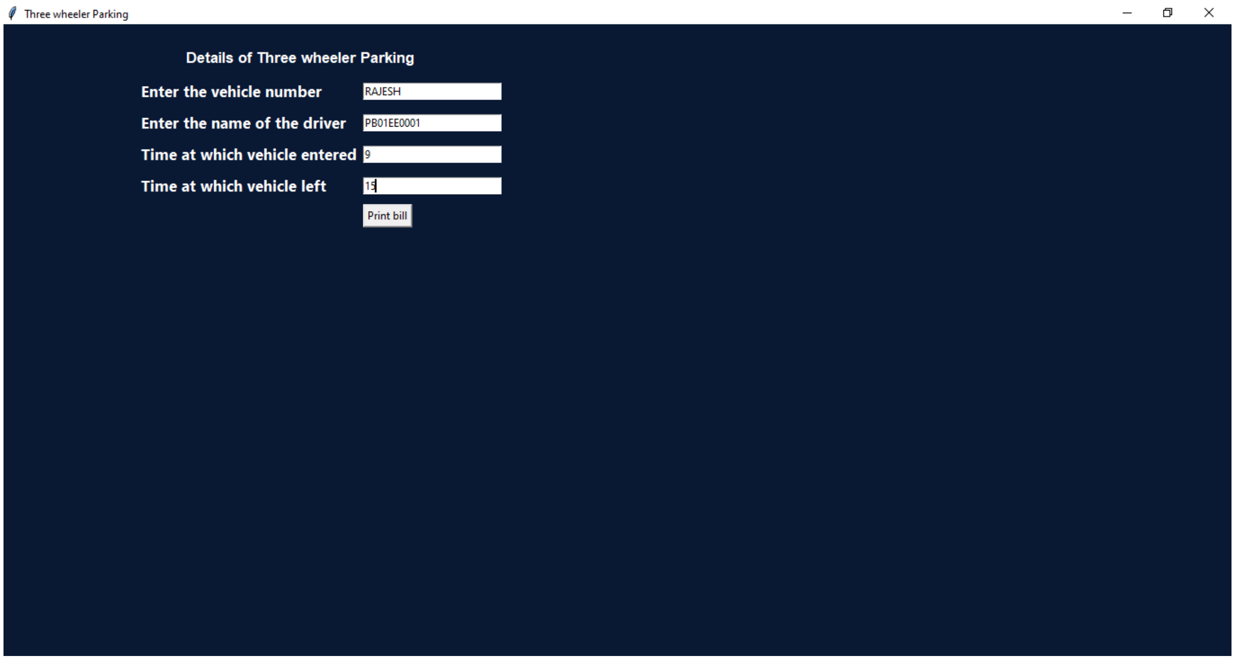
vehicle parking fee: 250.0

4. FILE CREATED TO STORE DETAILS:



```
vehicle registration number: TN20CM3224
vehicle driver name: SVGIRI
vehicle entering time: 12
vehicle leaving time: 14
Vehicle parking fee: 100.0
vehicle registration number: TN54CF1234
vehicle driver name: RAMESH
vehicle entering time: 10
vehicle leaving time: 15
Vehicle parking fee: 250.0
```

5.THREE-WHEELER (on clicking three wheeler button)



Details of Three wheeler Parking

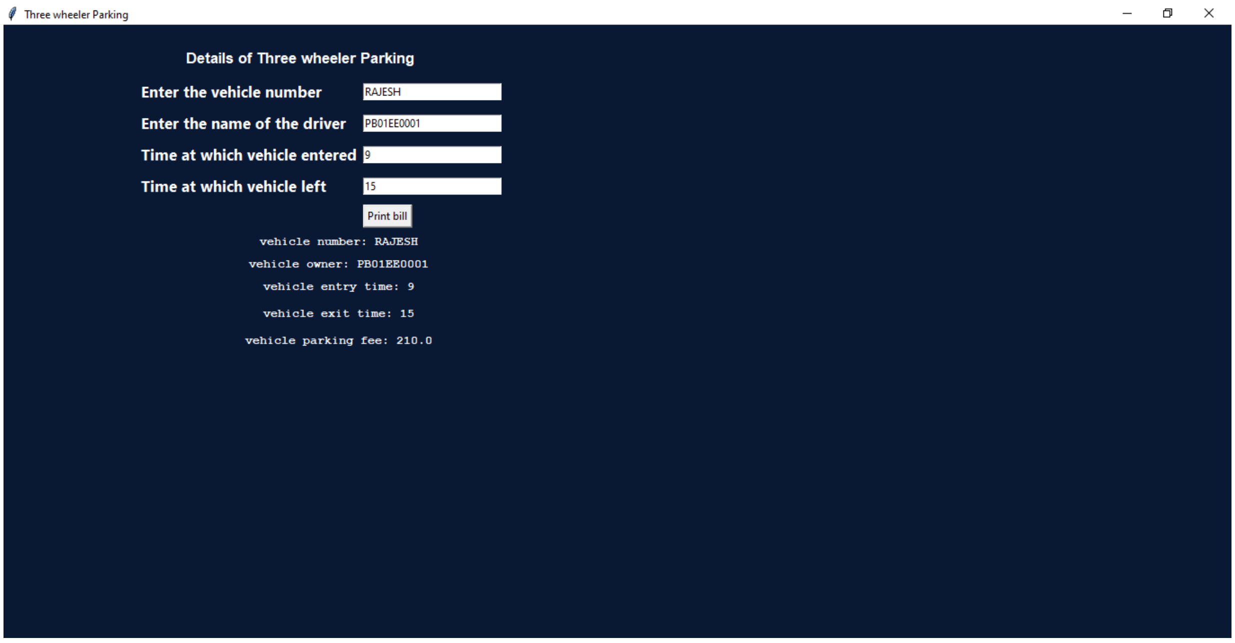
Enter the vehicle number RAJESH

Enter the name of the driver PB01EE0001

Time at which vehicle entered 9

Time at which vehicle left 15

6.PRINT BILL(for three-wheeler details)



Details of Three wheeler Parking

Enter the vehicle number RAJESH

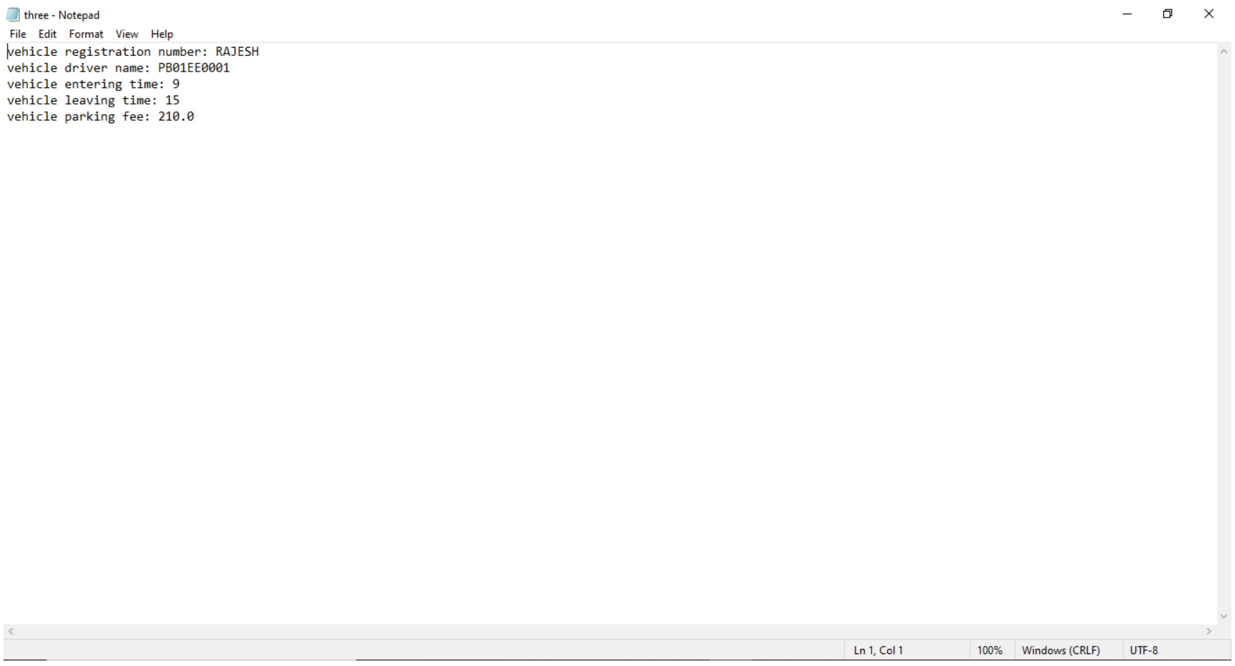
Enter the name of the driver PB01EE0001

Time at which vehicle entered 9

Time at which vehicle left 15

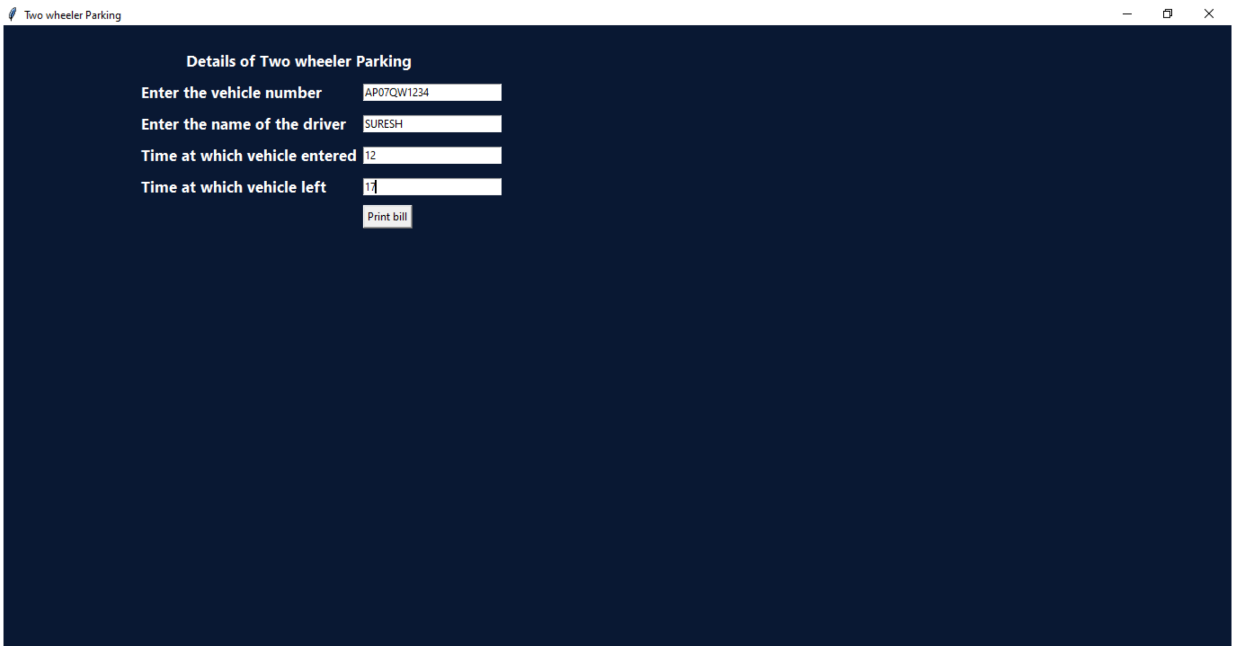
```
vehicle number: RAJESH
vehicle owner: PB01EE0001
vehicle entry time: 9
vehicle exit time: 15
vehicle parking fee: 210.0
```

7.FILE FOR THREE-WHEELER



```
vehicle registration number: RAJESH
vehicle driver name: PB01EE0001
vehicle entering time: 9
vehicle leaving time: 15
vehicle parking fee: 210.0
```

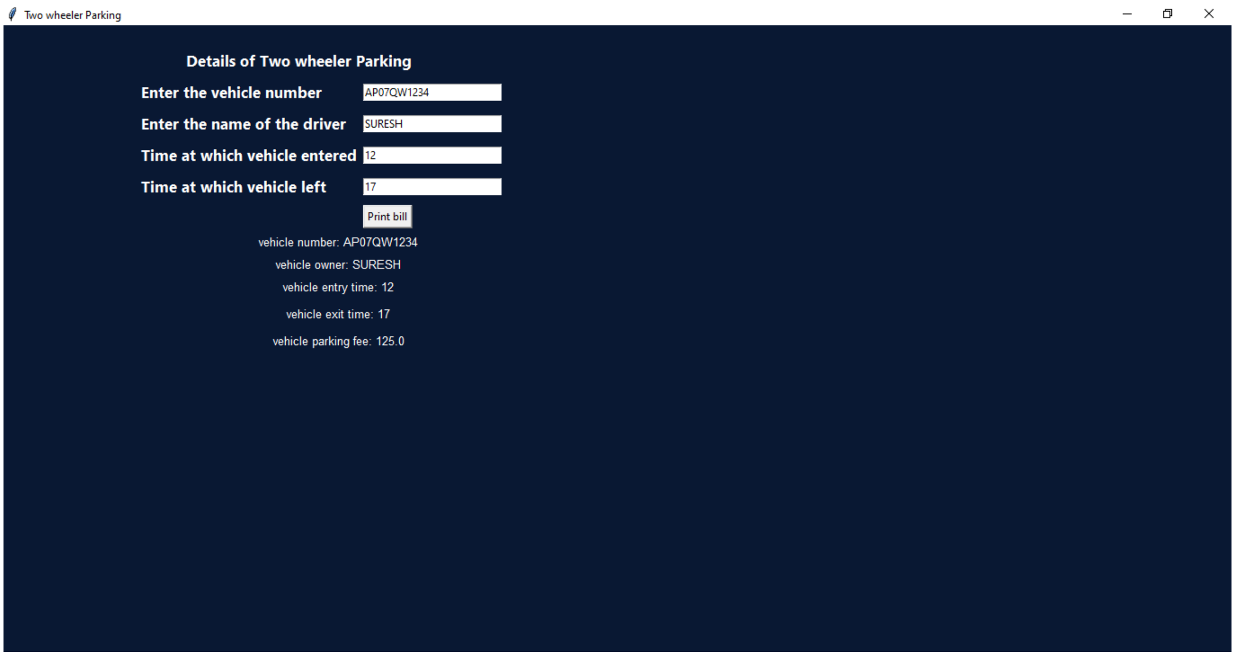
8.TWO WHEELER (on clicking button)



Details of Two wheeler Parking

Enter the vehicle number	AP07QW1234
Enter the name of the driver	SURESH
Time at which vehicle entered	12
Time at which vehicle left	17

9.PRINT BILL (on clicking button)



Details of Two wheeler Parking

Enter the vehicle number AP07QW1234

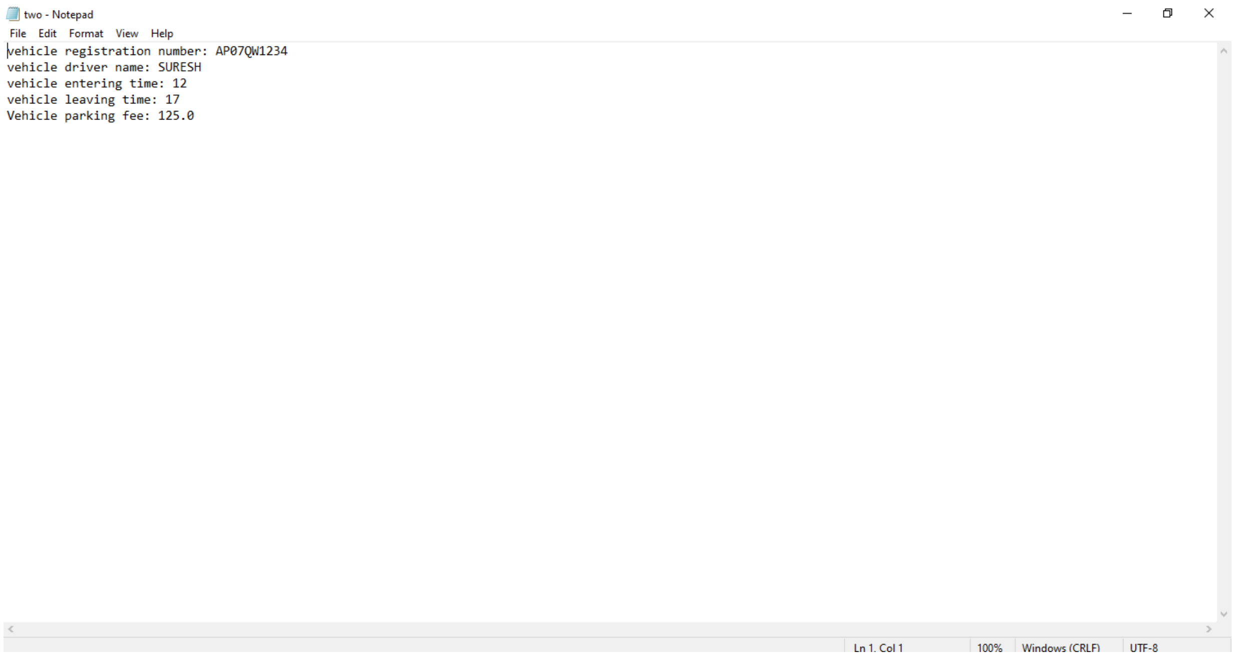
Enter the name of the driver SURESH

Time at which vehicle entered 12

Time at which vehicle left 17

```
vehicle number: AP07QW1234
vehicle owner: SURESH
vehicle entry time: 12
vehicle exit time: 17
vehicle parking fee: 125.0
```

10.FILE FOR TWO-WHEELER



```
vehicle registration number: AP07QW1234
vehicle driver name: SURESH
vehicle entering time: 12
vehicle leaving time: 17
Vehicle parking fee: 125.0
```

<u>RESULT :</u>

We finally got the end product as a “parking management system interface” that includes all the above-mentioned modules. We had learnt how to make a GUI using tkinter in python and learnt to store it in files.

In this project we can give details of the driver and enter the vehicle entry time and exit time. And with the given information we can generate parking fee and display to the driver

And all the records of two-wheeler, three-wheeler, and four wheeler are stored sequentially in the files along with parking fee .

Finally, by doing this project we learnt about GUI tkinter, functions, file handling in python.

REFERENCES:

- https://www.tutorialspoint.com
- https://stackoverflow.com
- https://www.w3schools.com
- https://www.geeksforgeeks.org