AKNOWLEDGEMENT

I am overwhelmed in all humbleness and gratefulness to acknowledge my depth to my parents, teachers and my teammate those who have helped me to do this project, well above the level of simplicity and into something concrete. I would like to express my special thanks of gratitude to my teacher who gave me the golden opportunity to do this wonderful project which also helped me in doing a lot of Research and I came to know about so many new things. I am really thankful to them. Any attempt at any level can‘t be satisfactorily completed without the support and guidance of my parents and friends. I would like to thank my parents and teammate who helped me a lot in gathering different information, collecting data and guiding me in making this project.

ABSTRACT

Digi tracker is a program which works with Tkinter for the GUI and threading for parallel execution of two different functions which enables the user to use the pause/resume button for infinitely many times. The program uses win32gui module to get the name of background application. It uses matplotlib for showing the data in graphical format. We have also used Tkfonts for custom fonts to get a better look. The buttons and background as mostly custom images which are imported and used for better interface. It also uses MySQL for backend to store the recorded time for each application.

PROJECT REQUIRMENTS

HARDWARE REQUIRMENTS:

- Minimum 4GB RAM
- Wi-Fi router
- CPU having a minimum of 5.2GHz clock speed

SOFTWARE REQUIRMENTS

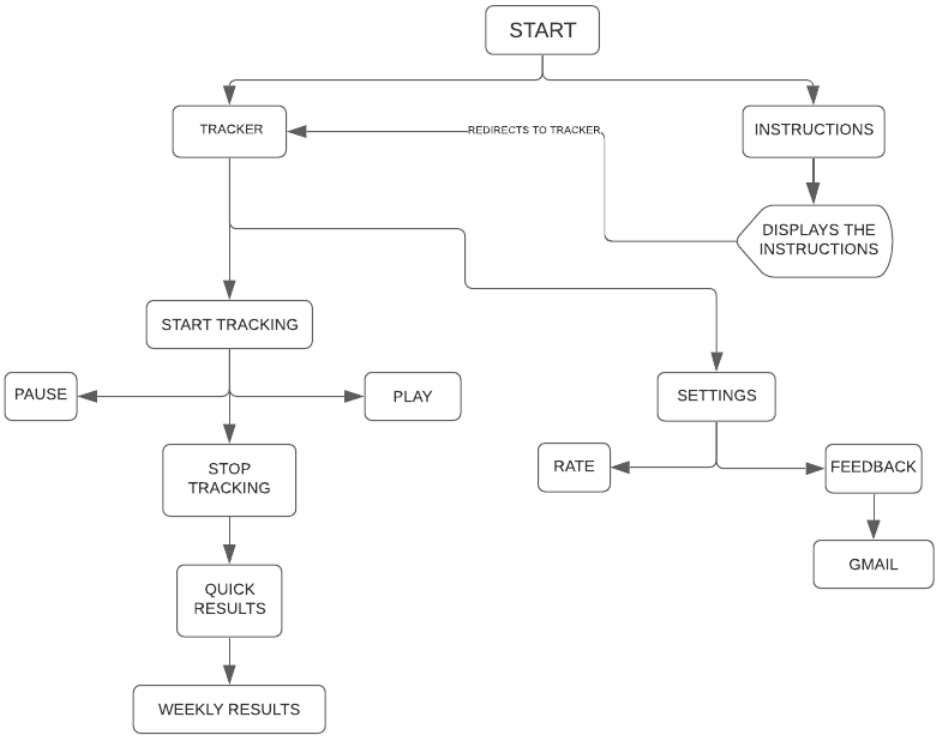
- PyCharm
- Python
- MySQL 8.0
- Chrome
- Python 3.9
- Packages used are win32gui, threading, matplotlib, webbrowser, psutil.

APPLICATIONS:

This program can be used by every person who wants to manage their time in a proper way and want to be more efficient. It can also be used by parents to track the activities of their kids when on screen. It can also be used by companies to track the work of the employees. This program can also be implemented in future in schools when digital learning takes over and education takes place in a single screen.

DESIGN

BLOCK DIAGRAM

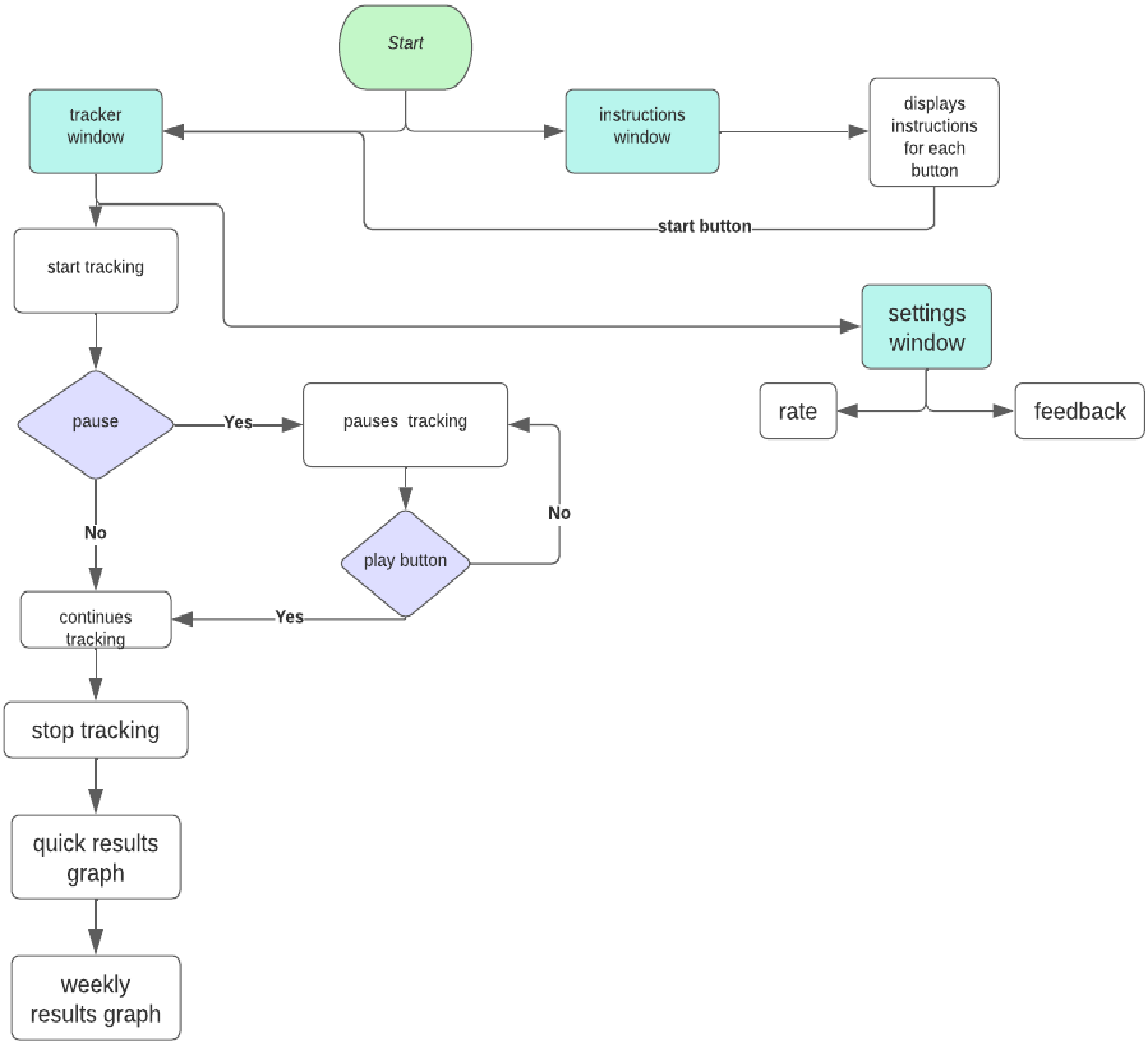


EXPLANATION

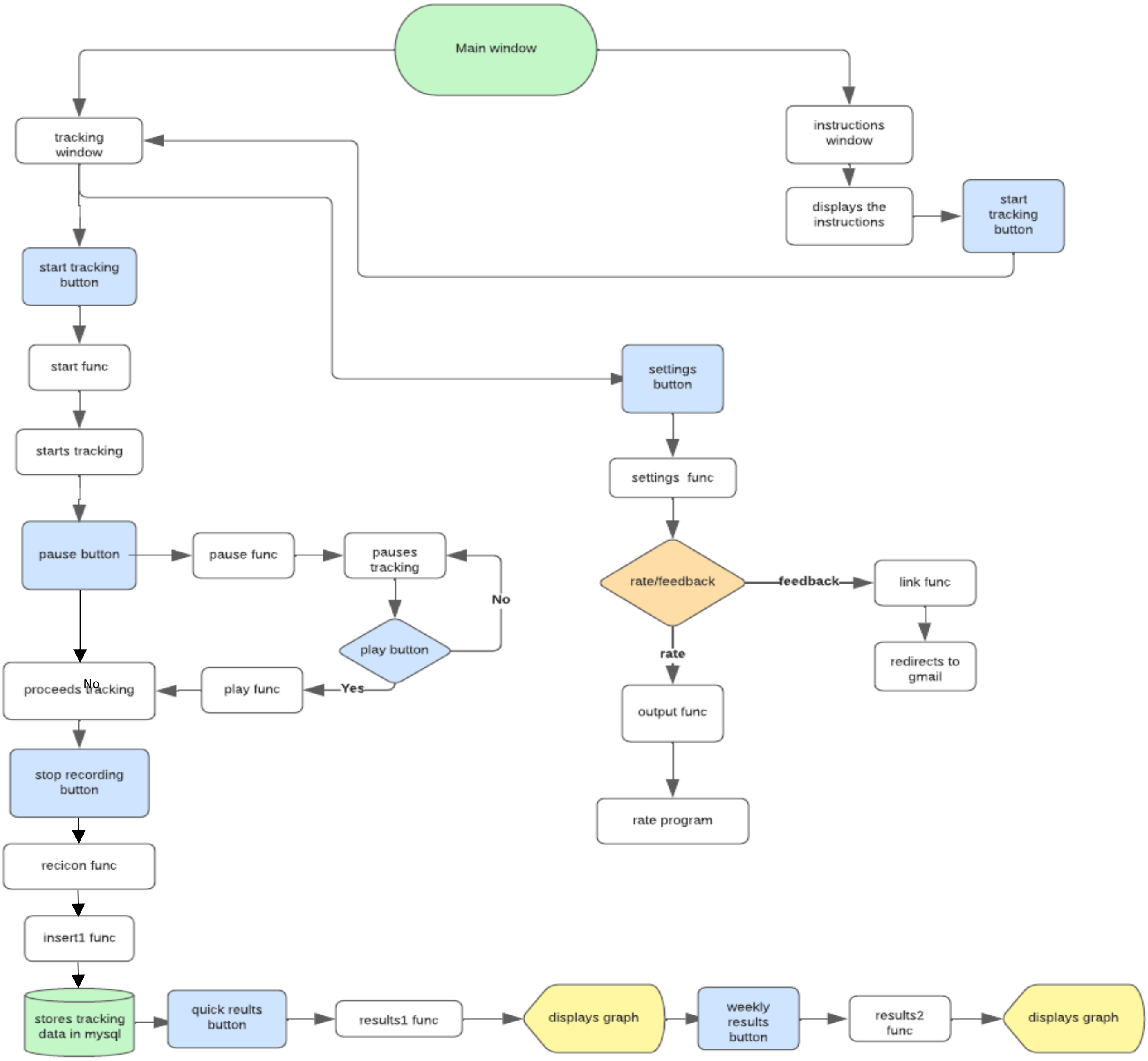
First we get the starting window that lets us choose whether we want instructions or to directly proceed to the tracker. In the instructions window, it displays the instructions for each button and its functions. There is also a button in the instruction window that redirects us to the tracker. In the tracker, we can start the tracking by clicking the start button and it will continue to track till we pause or stop the tracking. Once the tracking is over the quick results button is enabled which when clicked, shows a line graph on the time used on each application. The weekly results button is also enabled when the quick results button is clicked. The weekly results button, when clicked, displays the total usage of each day per day in a week. For every week the data of previous week is erased for the new week's data to be entered. There is also a settings button where you can send feedback directly send mail to us or rate us using the 'rate us' option.

FLOWCHART

A)Project flow



B)Program Flow



ALGORITHM

1. Start
2. We can choose one of the two modes which are
   (A) Instructions mode: It displays the instructions for each button available in the program
   (B) Start tracking mode: It directly shows the tracking window
3. Start tracking option in the tracking window initiates the tracking sequence which starts the tracker
4. Pause/play option enables us to pause or resume the tracking according to our wish
5. Stop recording option terminates the tracking sequence and the tracking data is stored in the database
6. View quick results option displays the quick graph for that session of using the tracker
7. View weekly results option displays the weekly graph for that week of using the tracker
8. Settings option opens a settings tab which allows you to rate or give feedback to the developer.

SOURCE CODE

MAIN PROGRAM

```
import mysql.connector as sql
mycon = sql.connect(host="localhost", user="root", passwd="nikjan09578", database="firstcodes")
import tkinter.font
import matplotlib.pyplot as plt
from matplotlib.figure import Figure
from tkinter.font import Font
from matplotlib.backends.backend_tkagg import FigureCanvasTkAgg,NavigationToolbar2Tk
from win32gui import GetForegroundWindow
import time
import psutil
import win32process
import threading
from tkinter import *
import collections,functools,operator
from backend import *
from PIL import *
from tkinter import messagebox
import webbrowser as wb
from pandas import DataFrame
root=Tk()
root.geometry("1000x650")
root.title("DIGI TRACKER")
root.config()
root.iconbitmap('A:/icon.ico')
```

```
#nf=tkinter.font.Font(family='Times')



bimg=PhotoImage(file="A:/bgimg.png")


t_=Label(root,image=bimg)

t_.place(x=0,y=0)


t=Label(root,text='Welcome to Digi-Tracker',font='times 30 bold
italic',bg='#0a104a',fg='yellow')

t.place(x=280,y=20)


#font1= tkinter.font.Font(size=20)

t1=Label(root, text='Existing user ?....click below ↓
',font='helvetica',bg='#081041',fg='gold')

t1.config(font=(13))

t1.place(x=370,y=225)


b1=Button(root,font='helvetica 20',text='Click to start',height=1, justify=CENTER,

        borderwidth=2,relief='raised',bg='orange',fg='white',command=lambda:[start()])

b1.place(x=410,y=250)


t2=Label(root, text='New user ?....click below ↓ ',font='helvetica',bg='#031032',fg='gold')

t2.config(font=(13))

t2.place(x=386,y=340)


b2=Button(root,font='helvetica 20',text='Click for instructions',height=1, justify=CENTER,

        borderwidth=2,relief='raised',bg='orange',fg='white',command=lambda:[instructions()])

b2.place(x=373,y=365)
```

```
#IMAGES

pausebtn=PhotoImage(file="A:/pause-icon.png")

playbtn=PhotoImage(file="A:/play-icon.png")

bimg2=PhotoImage(file="A:/bgimg2.png")

bgimg3=PhotoImage(file="A:/bgimg3.png")

rec=PhotoImage(file='A:/recording-icon.png')

st=PhotoImage(file='A:/start-icon.png')

icon=PhotoImage(file='A:/icon.png')

settings=PhotoImage(file='A:/settings_icon.png')



var=True



r=[]    #stores the list of dictionaries to be added fully......see new.py to know how too add the values of dictionaries with same keys


def run():

    while True:

        times = {}

        timestamp = {}

        currentapp = psutil.Process(win32process.GetWindowThreadProcessId(GetForegroundWindow())[1]).name().replace(".exe" , "")

        currentapp=str(currentapp)

        currentapp=currentapp.strip()

        timestamp[currentapp] = int(time.time())

        time.sleep(10)
```

```
          if currentapp not in times.keys():
                        times[currentapp] = 0
          times[currentapp] = (times[currentapp]+int(time.time())-timestamp[currentapp])
          global r
          r.append(times.copy())
          if var==False:
              break
        # do not add the dictionary directly.....



def true():
    global var
    var = True
    threading.Thread(target=run).start()




def false():
    global var
    var = False
    print(r)


def calc():
    global g
    g=dict(functools.reduce(operator.add,map(collections.Counter,r)))
    g=g.items()
    global g1
    g1=list(g)
    print(g1)
```

```
#CUSTOM FONTS
a=Font(family='Elephant', size=17)
b=Font(family='Stencil', size=20)



def start():
    global new
    new=Toplevel(root)
    new.title('DIGI TRACKER')
    new.geometry("1000x650")
    new['bg']='RoyalBlue3'
    new.iconbitmap('A:/icon.ico')
    k=Label(new,image=bimg)
    k.place(x=0,y=0)
    k1=Label(new,
            text='Tracker',
            font='times 20 bold',
            image=bimg,
            relief='ridge',
            borderwidth=3,
            bg='aquamarine',
            width=523,
            height=345,
            anchor='nw',
            fg='blue')
    k1.place(x=1,y=0)

    k2=Label(new,
            text='CLICK TO START TRACKING',
         font=a,
         bg='#17107c',
         fg='lawn green',
         justify='left')
k2.place(x=20,y=100)

q=Label(new,
        text='🔻',
        bg='#17107c',
        fg='lawn green',
        font=a,
        )
q.place(x=110,y=150)
k3=Label(new,
         text='View Results',
         image=bimg2,
         font='Helvetica 20',
         relief='ridge',
         borderwidth=3,
         width=465,
         height=345,
         anchor='nw',
         bg='aquamarine',
         fg='gold')
k3.place(x=530)

k6=Label(new,
         text='weekly results',
         image=bgimg3,
         font='times 20 bold',
         relief='ridge',
         borderwidth=3,
         bg='aquamarine',
         fg='gold',
         width=993,
         height=293,
         anchor='nw')
k6.place(x=1,y=350)

def start():
    global n1
    n1=Button(new,
              image=st,
              font='helvetica 30',
              borderwidth=0,
              bg='#120f6e',
              command=lambda:[recicon(),pause(),pausetext(),true(),k2.destroy(),q,destroy()])
    n1.place(x=100,y=220)

start()

def notis():
         messagebox.showinfo('digi tracker',"tracking terminated successfully!!")



def play():
    global n3
    n3= Button(new,image=playbtn, font='helvetica 30',bg='#0d1056', borderwidth=0,
                command=lambda:[pause(),pausetext(),true(),k4.destroy()])
        n3.place(x=250, y=220)

    def pause():
        global n2
        n2 = Button(new, image=pausebtn, font='helvetica 30',bg='#0d1056',
                 borderwidth=0, command=lambda: [play(),playtext(),false()])
        n2.place(x=250, y=220)
        #n1.destroy()



    def playtext():
        global k4
        k4 = Label(new, text='TRACKING PAUSED....', font=a,
                bg='#17107c',fg='lawn green')
        k4.place(x=40, y=100)

    def pausetext():
        k5=Label(new,text='CLICK TO PAUSE....',font=a,
              bg='#17107c',fg='lawn green')
        k5.place(x=40,y=100)
# stop_rec_icon

    def recicon():
        global n4
        n4=Button(new,image=rec,font='helvetica 30',borderwidth=0,
              bg='#0b104e',command=lambda:[false(),notis(),calc(), enable(),insert1(g1)])
        n4.place(x=380,y=220)
    def enable():
        n5['state'] = 'active'

    def enable2():
        n6['state']= 'active'

    def link():
        wb.open_new_tab('A:/home.html')

    def dropdown():
            global new3,a1,a2,a3,a4,a5,a6,c1,c3,c4,c5,c6
            new3=Toplevel(new)
            new3.geometry('200x150')
            new3.iconbitmap('A:/icon.ico')

            l=Label(new3,image=bgimg3)
            l.place(x=0,y=0)

            c=Label(new3,text="Give us your feedback by\nclicking this button",fg='lawn green',bg="#110d44",font='helvetica 12')
            c.place(x=15,y=3)

            b1 = Button(new3, text='CLICK HERE', borderwidth=3, relief='ridge',fg='lawn green',font='helvetica 10',bg="#501050" ,command=link)
            b1.place(x=56, y=50)

            a1=Label(new3,text='RATE OUR PROGRAM!!',fg='lawn green',bg="#110d44",font='helvetica 9')
            a1.place(x=35,y=87)
```

```
a2=Label(new3,text='1',fg='lawn green',bg="#110d44",font='helvetica 11')
a2.place(x=45,y=130)

c1=Radiobutton(new3,bg="#110d44",command=output)
c1.place(x=45,y=110)

a3=Label(new3,text='2',fg='lawn green',bg="#110d44",font='helvetica 11')
a3.place(x=65,y=130)

c3=Radiobutton(new3,bg="#110d44",command=output)
c3.place(x=65,y=110)

a4=Label(new3,text='3',fg='lawn green',bg="#110d44",font='helvetica 11')
a4.place(x=85,y=130)

c4=Radiobutton(new3,bg="#110d44",command=output)
c4.place(x=85,y=110)

a5=Label(new3,text='4',fg='lawn green',bg="#110d44",font='helvetica 11')
a5.place(x=105,y=130)

c5=Radiobutton(new3,bg="#110d44",command=output)
c5.place(x=105,y=110)

a6=Label(new3, text='5',fg='lawn green',bg="#110d44",font='helvetica 11')
a6.place(x=125, y=130)

c6=Radiobutton(new3,bg="#110d44",command=output)
```

```
            c6.place(x=125,y=110)



    def setting():
        x=Button(new,image=settings,borderwidth=0,bg='#0b104e',command=dropdown)
        x.place(x=7,y=605)
    setting()

    def output():
        q1=Label(new3,text="THANK YOU\nFOR RATING US!!!",bg="#110d44",fg='lawn green',font='helvetica 11')
        q1.place(x=35,y=90)
        a1.destroy()
        a2.destroy()
        a3.destroy()
        a4.destroy()
        a5.destroy()
        a6.destroy()
        c1.destroy()
        c3.destroy()
        c4.destroy()
        c5.destroy()
        c6.destroy()

        #use radial buttons and a confirmation button for rating



    def results1():
        global n5

        k32=Label(new,text='View
Results',font=a,borderwidth=0,relief='ridge',bg='#06103a',fg='lawn green')

        k32.place(x=540,y=10)

        n5=Button(new,text="click to view results", font=b, borderwidth=0,

                  state=DISABLED,relief='ridge',
command=lambda:[plot(),enable2(),n5.destroy()])

        n5.place(x=600,y=90)

    results1()


    def results2():

        global n6

        n6=Button(new,text="click to view weekly results",font=b, borderwidth=0,

                  state=DISABLED,relief='ridge', command=lambda:[plot2(),n6.destroy()])

        n6.place(x=300,y=400)

    results2()

#   TOP LEFT GRAPHING


    cursor = mycon.cursor()


    def plot():


        global df1

        q = 'select * from project'

        cursor.execute(q)

        row = cursor.fetchall()

        row = dict(row)


        dk = row.keys()

        dk = list(dk)
        dv = row.values()
        dv = list(dv)

        f = plt.Figure(figsize=(4, 3), dpi=100)
        f.add_subplot(111).plot(dk, dv)

        chart = FigureCanvasTkAgg(f, new)
        chart.get_tk_widget().place(x=550, y=45)

    #create the big graph in the down position for the total time data[CHANGE TO BAR GRAPH

    def plot2():
        m="select * from weekly_use"
        cursor.execute(m)
        row2=cursor.fetchall()
        row2= dict(row2)

        dk1=row2.keys()
        dk1=list(dk1)

        dv1=row2.values()
        dv1=list(dv1)

        f1=plt.Figure(figsize=(7,3),dpi=100)
        f1.add_subplot(111).plot(dk1, dv1)

        chart2=FigureCanvasTkAgg(f1,new)
        chart2.get_tk_widget().place(x=200,y=353)

    new.mainloop()






def instructions():
    global new2
    new2=Toplevel(root)
    new2.title('INSTRUCTIONS')
    new2.geometry("1000x650")
    new2.iconbitmap('A:/icon.ico')

    b = Font(family='Stencil', size=15)

    i1=Label(new2,image=bimg)
    i1.place(x=0,y=0)

    i2=Button(new2,image=st,borderwidth=0,bg='#17107b')
    i2.place(x=30.,y=85)

    i3=Button(new2,image=pausebtn,borderwidth=0,bg='#140982')
    i3.place(x=30,y=175)                              # have 80 pixel gap b/w 2 adjacent buttons

    i4=Button(new2,image=playbtn,borderwidth=0,bg='#1d0f8a')
    i4.place(x=30,y=265)
    i5=Button(new2,image=rec,borderwidth=0,bg='#0e105c')
    i5.place(x=30,y=355)

    i6=Button(new2,text="click to view results", font=b, borderwidth=0,relief='ridge')
    i6.place(x=30,y=443)

    i7=Button(new2,text="click to view weekly results",font=b, borderwidth=0,relief='ridge')
    i7.place(x=15,y=505)

    i8=Button(new2,image=settings,borderwidth=0,bg='#210e47')
    i8.place(x=35,y=563)

    i9=Button(new2,text='START',font=b,relief='ridge',borderwidth=2,command=
lambda:[start()])
    i9.place(x=900,y=597)



    l1=Label(new2,text="HOW TO USE DIGI TRACKER",
             font=a,
             fg='spring green',
             bg='#131070')
    l1.place(x=300,y=5)


    l2=Label(new2,
             text='This is the start button, this button is available in the "tracker" section which
allows you to initiate the tracking\nsequence'
                  ' to keep track of your app usage.Clicking this button will start the tracking. ',
             font='helvetica 11',
             justify='left',
             bg='#0a104d',
          fg='lawn green')
    l2.place(x=125,y=90)

    l3 = Label(new2,
            text='This is the pause button, this button allows you to pause the tracking.This button
is also available in the\n"tracker" section '
                  'and appears when you have initiated the tracking .i.e. when you clicked the
start button.',
            font='helvetica 11',
            justify='left',
            fg='lawn green',
            bg='#0a0f4d')
    l3.place(x=125,y=185)

    l4 = Label(new2,
            text='This is the play button,this button resumes your tracking process which was
paused using the pause button. This button\nis also available'
                  ' in the "tracker" section and it appears in the same position as that of the pause
button. ',
            font='helvetica 11',
            justify='left',
            fg='lawn green',
            bg='#051038')
    l4.place(x=125,y=278)

    l5 = Label(new2,
            text='This is the stop-recording button, this button is enabled as you click the start
button. This button\nis also available in '
                  'the "tracker" section which can be used to terminate the tracking sequence.',
            font='helvetica 11',
            justify='left',
            fg='lawn green',
            bg='#050d3c')
    l5.place(x=125,y=366)

    l6 = Label(new2,
            text='This is the quick analysis or result button which displays the usage of the applications in a graphical format.\nThis button is available in the '
                    '"view results" section and it is enabled when you click the stop recoding button.  ',
            font='helvetica 11',
            justify='left',
            fg='lawn green',
            bg='#040f35')
    l6.place(x=290,y=443)

    l7 = Label(new2,
            text='This is the weekly analysis button, it displays the overall usage of a week in a graphical format.This\nbutton is available in the '
                    '"weekly results" section, it is enabled when you click the quick results button ',
            font='helvetica 11',
            justify='left',
            fg='lawn green',
            bg='#04102e')
    l7.place(x=340,y=505)

    l8 = Label(new2,
            text='This is the settings button. When clicked,it shows a list of options(review and troubleshooting) where you can choose to\neither rate '
                    'our program or to develop our program by highlighting the bugs to our email id present in the troubleshooting option',
                font='helvetica 11',
                justify='left',
                fg='lawn green',
                bg='#071030')
    l8.place(x=80,y=563)

    l9=Label(new2,
             text='Click to start!!!!--→',
             font='helvetica 12',
             fg='lawn green',
             bg='#120731')
    l9.place(x=753,y=617)




    new2.mainloop()



root.mainloop()
```

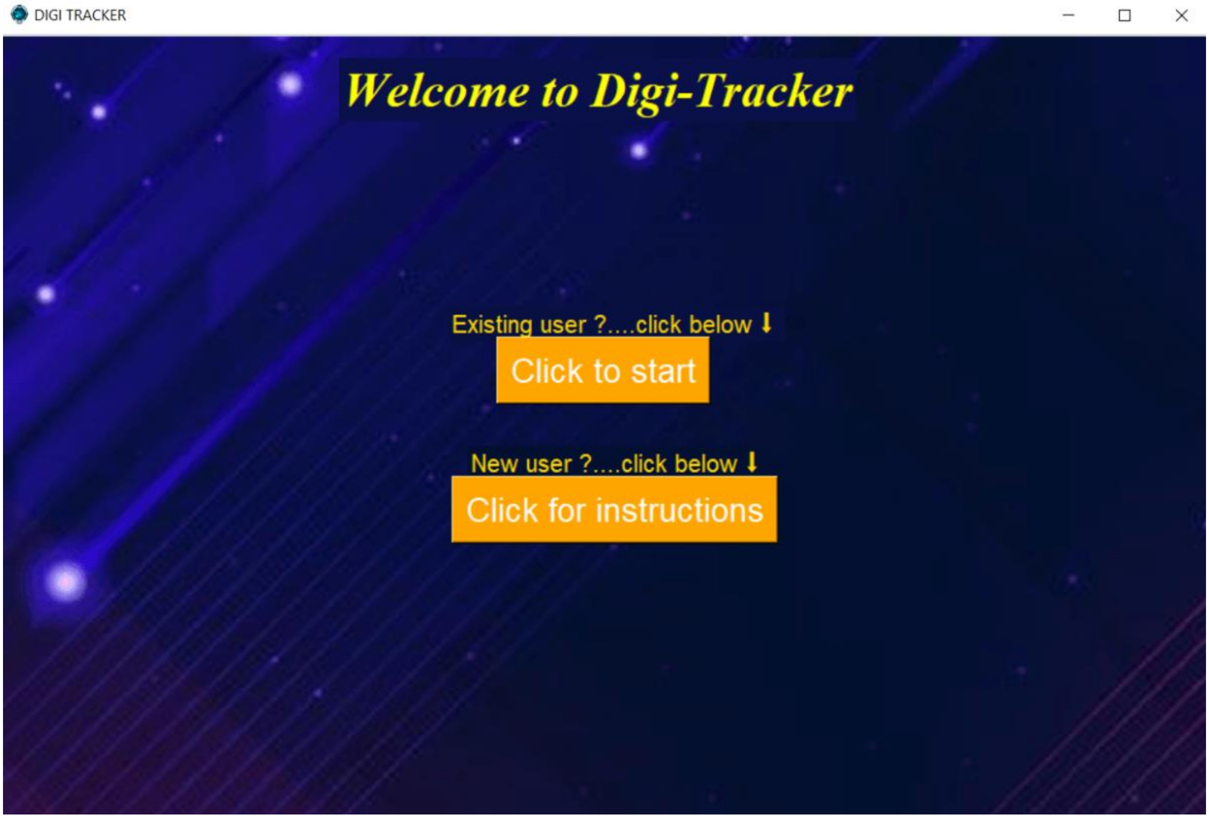
BACKEND MYSQL CONNECTOR

```
def insert1(lt):

 import mysql.connector as sql

 mycon = sql.connect(host="localhost", user="root", passwd="nikjan09578",
database="firstcodes")
 if mycon.is_connected() == False:
   print("error")



 cursor = mycon.cursor()
 e="delete from project"
 cursor.execute(e)

 for i in lt:
         print(i[0],i[1])
         a="select count(*) from project where app='{}'".format(i[0])
         cursor.execute(a)
         row=cursor.fetchone()

         if row[0] == 0:
                 q="insert into project values('{}',{})".format(i[0],i[1])
             cursor.execute(q)
             mycon.commit()

         if row[0] != 0:
```

```
        r="update project set time={} where app='{}'".format(i[1],i[0])
        cursor.execute(r)
        mycon.commit()



a='update project set time=time/60'
cursor.execute(a)
mycon.commit()
cursor=mycon.cursor()
c1="select count(*) from weekly_use"
cursor.execute(c1)
s1=cursor.fetchone()

if s1[0]==7:
   c2="delete from weekly_use"
   cursor.execute(c2)
   mycon.commit()

   c3="select dayname(sysdate())"
   cursor.execute(c3)
   s3=cursor.fetchone()
   v1=s3[0]     #dayname is stored(string)


   c4="select sum(time) from project"
   cursor.execute(c4)
   s4=cursor.fetchone()
   v2=s4[0]  #total time for the day is stored(floating point number)

   c5="insert into weekly_use values('{}',{})".format(v1,v2)
```

```
    cursor.execute(c5)
    mycon.commit()



 else:
    c3 = "select dayname(sysdate())"
    cursor.execute(c3)
    s3 = cursor.fetchone()
    v1 = s3[0]  # dayname is stored

    c4 = "select sum(time) from project"
    cursor.execute(c4)
    s4 = cursor.fetchone()
    v2 = s4[0]  # total time for the day is stored

    c5 = "insert into weekly_use values('{}',{})".format(v1, v2)
    cursor.execute(c5)
    mycon.commit()

if __name__=='__main__':
    insert1(lt)
```

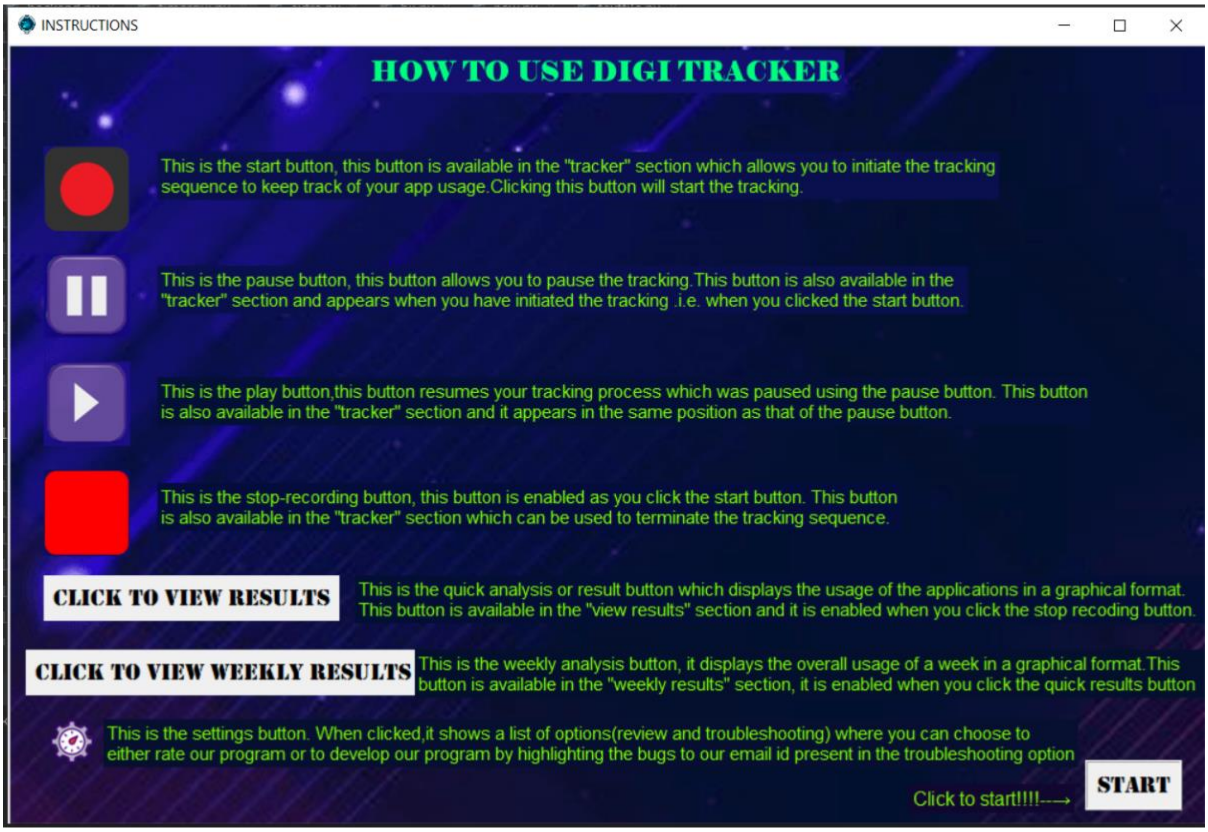
SAMPLE OUTPUT



Welcome to Digi-Tracker

Existing user ?....click below ⬇

New user ?....click below ⬇



HOW TO USE DIGI TRACKER

This is the start button, this button is available in the "tracker" section which allows you to initiate the tracking sequence to keep track of your app usage.Clicking this button will start the tracking.

This is the pause button, this button allows you to pause the tracking.This button is also available in the "tracker" section and appears when you have initiated the tracking .i.e. when you clicked the start button.

This is the play button,this button resumes your tracking process which was paused using the pause button. This button is also available in the "tracker" section and it appears in the same position as that of the pause button.

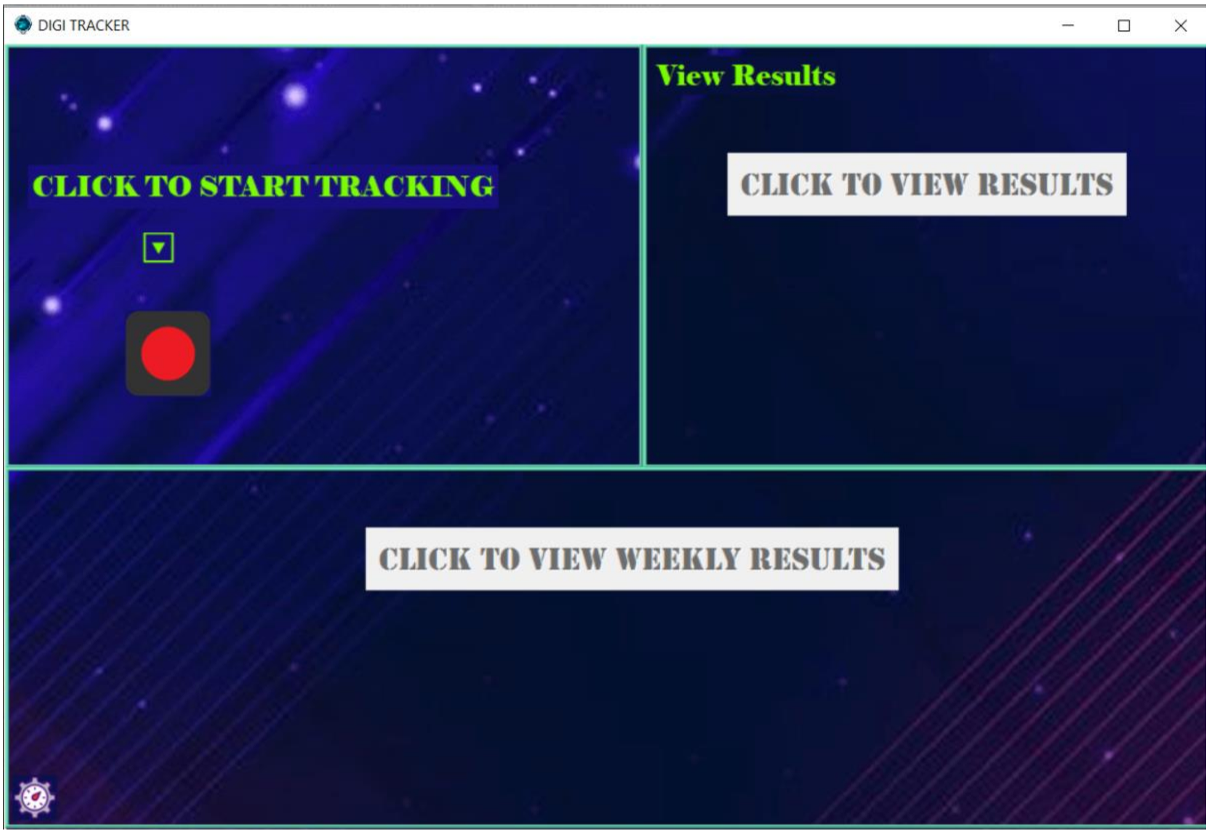
This is the stop-recording button, this button is enabled as you click the start button. This button is also available in the "tracker" section which can be used to terminate the tracking sequence.

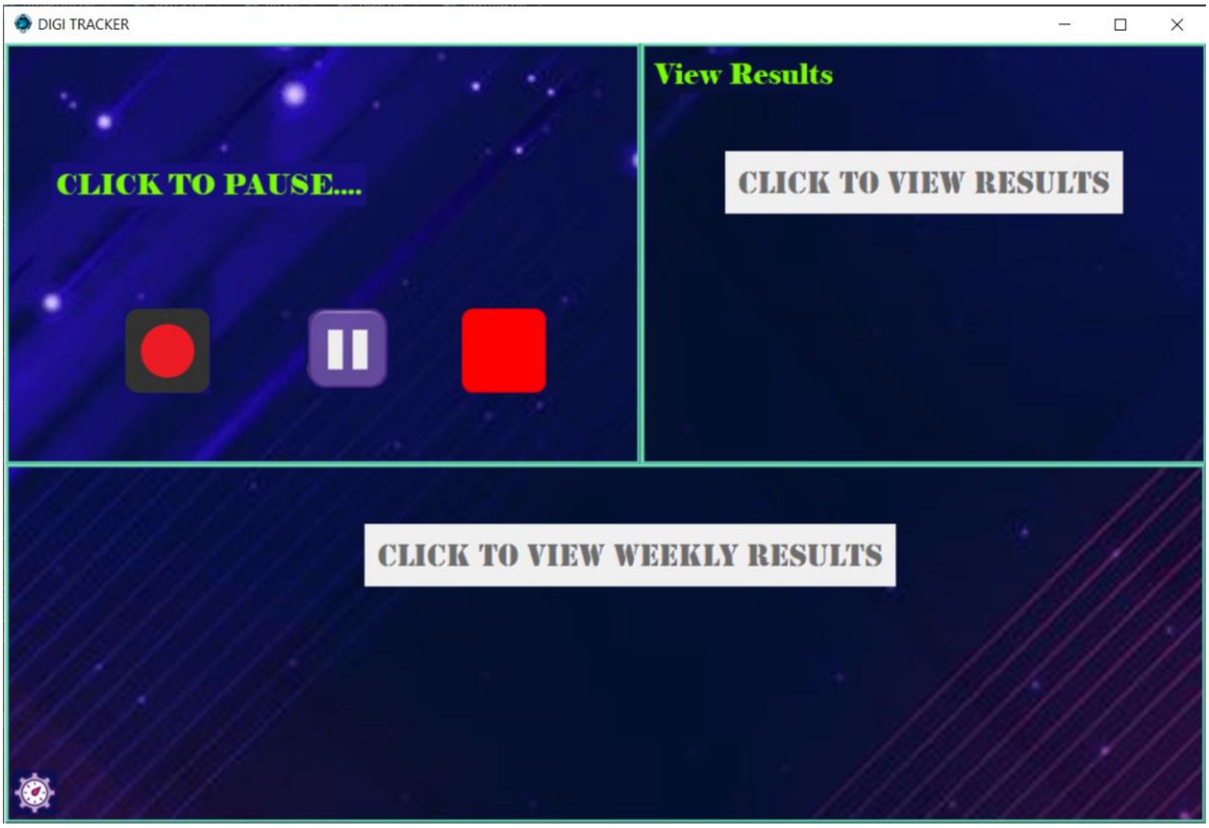
This is the quick analysis or result button which displays the usage of the applications in a graphical format. This button is available in the "view results" section and it is enabled when you click the stop recoding button.

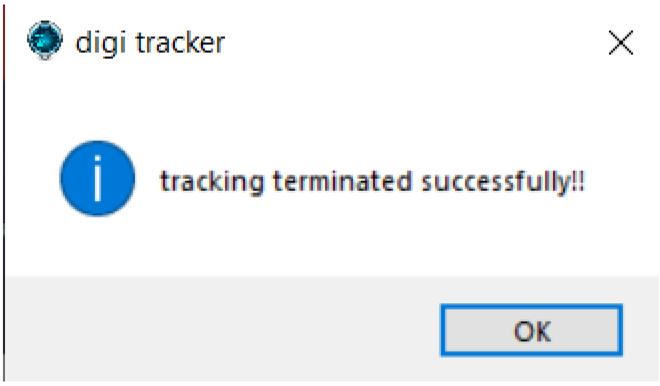
This is the weekly analysis button, it displays the overall usage of a week in a graphical format.This button is available in the "weekly results" section, it is enabled when you click the quick results button

This is the settings button. When clicked,it shows a list of options(review and troubleshooting) where you can choose to either rate our program or to develop our program by highlighting the bugs to our email id present in the troubleshooting option

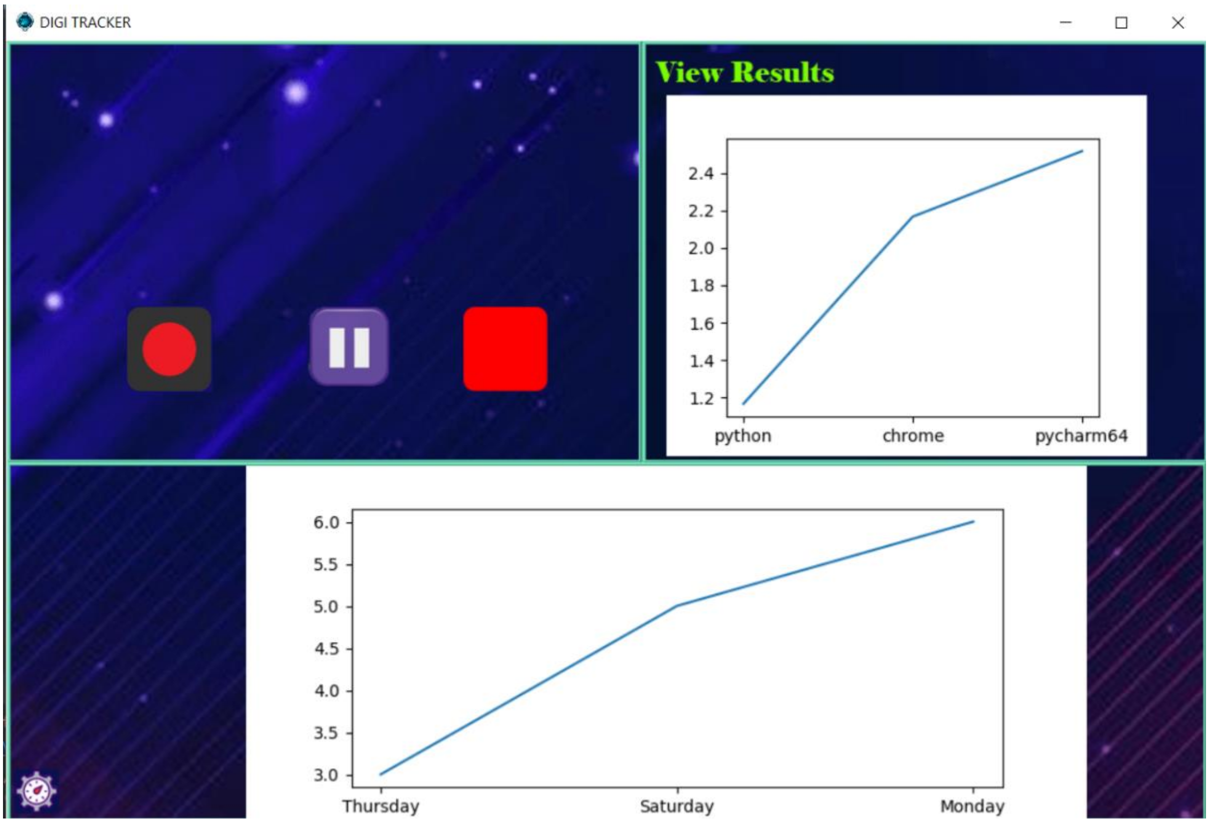
Click to start!!!!---→

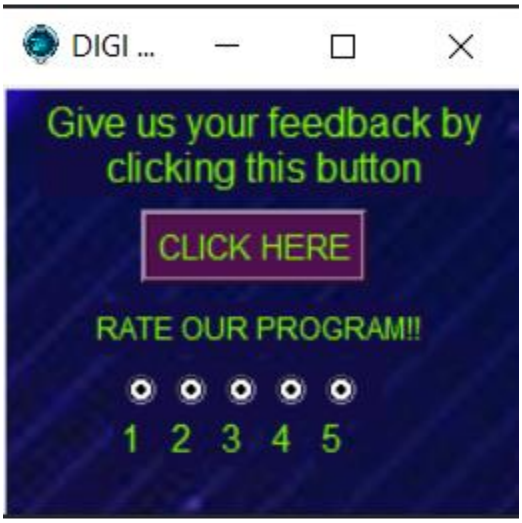

DIGI TRACKER
CLICK TO START TRACKING
View Results
CLICK TO VIEW RESULTS
CLICK TO VIEW WEEKLY RESULTS


DIGI TRACKER
CLICK TO PAUSE....
View Results
CLICK TO VIEW RESULTS
CLICK TO VIEW WEEKLY RESULTS



tracking terminated successfully!!





Give us your feedback by clicking this button

FUTURE ENHANCEMENTS

- We will try to extend the tracking even for individual websites
- Making the graphs better
- More accurate tracking (by eliminating the error in calculating time by few milliseconds)
- Adding an option to view weekly results before using the tracker
- Removing repetition of total time of each day if the tracker is used more than once in a day

BIBILIOGRAPHY:

- Stackoverflow
- GeekForGeeks
- Sumita Arora
- GitHub
- Codemy.com