MINI PROJECT REPORT

on

PROMPTER APP using Python

Submitted by

ADITYA KOMATH (RA2311004010370)
SHAIK ASIMA MUBIAS (RA2311004010388)

Semester – II

Academic Year: 2023-24 Even

Under the guidance of

Dr. Ananya Parameswaran
Assistant Professor, Department of ECE

In partial fulfilment for the Course

of

21CSS101J -PROGRAMMING FOR PROBLEM SOLVING

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
College of Engineering and Technology,
SRM Institute of Science and Technology
SRM Nagar, Kattankulathur – 603203, Kancheepuram District, Tamil Nadu.

April 2024

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

(Under Section 3 of UGC Act, 1956)

BONAFIDE CERTIFICATE

rtified that this activity report for the course 21CSS101J -PROGRAMMING FOR PROBLEM SOLVING is

bonafide work of **Aditya Komath (RA2311004010370) and Shaik Asima Mubias (RA2311004010388)**

o carried out the work under my supervision.

03/05/2024

SIGNATURE

Dr. Ananya Parameswaran
Assistant Professor
Department of ECE
SRMIST
Kattankulathur

SIGNATURE

Academic Coordinator
Department of ECE
SRMIST
Kattankulathur

TABLE OF CONTENTS

ABSTRACT

The Task Prompter is a Python-based program designed to assist users in managing their tasks efficiently by issuing regular reminders through pop-up notifications. The primary objective of the program is to help users stay organized by reminding them of specific tasks at predefined times.

The program allows users to input the details of the task they wish to be reminded of, including the task description and the time at which they want to receive the prompt. Once configured, the Task Prompter operates in the background, periodically checking the system clock to trigger notifications according to the specified schedule.

Users have the flexibility to snooze notifications, enabling them to temporarily delay reminders if necessary. The snooze functionality allows users to customize the duration of the delay, providing them with the option to postpone notifications for a suitable amount of time.

Implemented using Python, the Task Prompter utilizes libraries such as Tkinter for the graphical user interface (GUI) to facilitate user interaction and the scheduling module for managing reminders. The program ensures a user-friendly experience, with intuitive input mechanisms and clear notification prompts.

OBJECTIVE

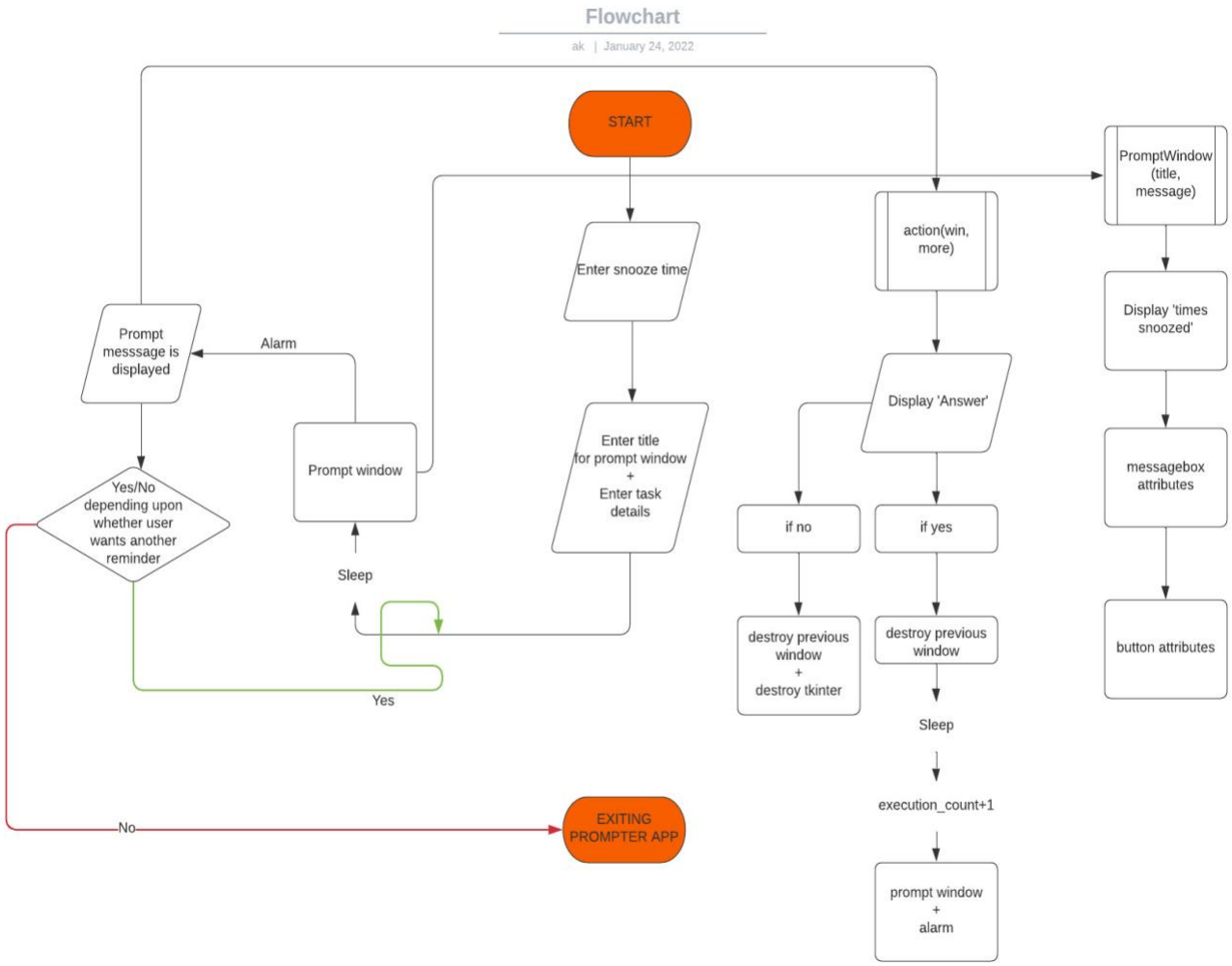
The purpose of the prompter program is to keep track of the task to do by issuing regularly spaced reminders of said task in the form of pop-up notifications. The user is required to enter the kind of task they want to be reminded of and at what time they will receive the prompt. The user can also choose to snooze the notification how many ever times they want.

INTRODUCTION

Remembering the tasks for a day can often prove to be very difficult. What makes it even more so is finding time for that task especially if your schedule is a congested one. A prompter might help in eliminating some of these common problems many of us face in our daily lives and assist in efficient time management.

Creating a program to keep track of the task to do by issuing regularly spaced reminders in the form of pop-up notifications will mitigate these issues and help us organize our day. The user is can enter the time in which they want to be reminded of their task. The user can also choose to snooze the notification how many ever times they want.

FLOWCHART


Flowchart
ak | January 24, 2022
START
Enter snooze time
Enter title for prompt window + Enter task details
Prompt window
Sleep
Prompt messsage is displayed
Alarm
Yes/No depending upon whether user wants another reminder
Yes
No
EXITING PROMPTER APP
action(win, more)
Display 'Answer'
if no
if yes
destroy previous window + destroy tkinter
destroy previous window
Sleep
execution_count+1
prompt window + alarm
PromptWindow (title, message)
Display 'times snoozed'
messagebox attributes
button attributes

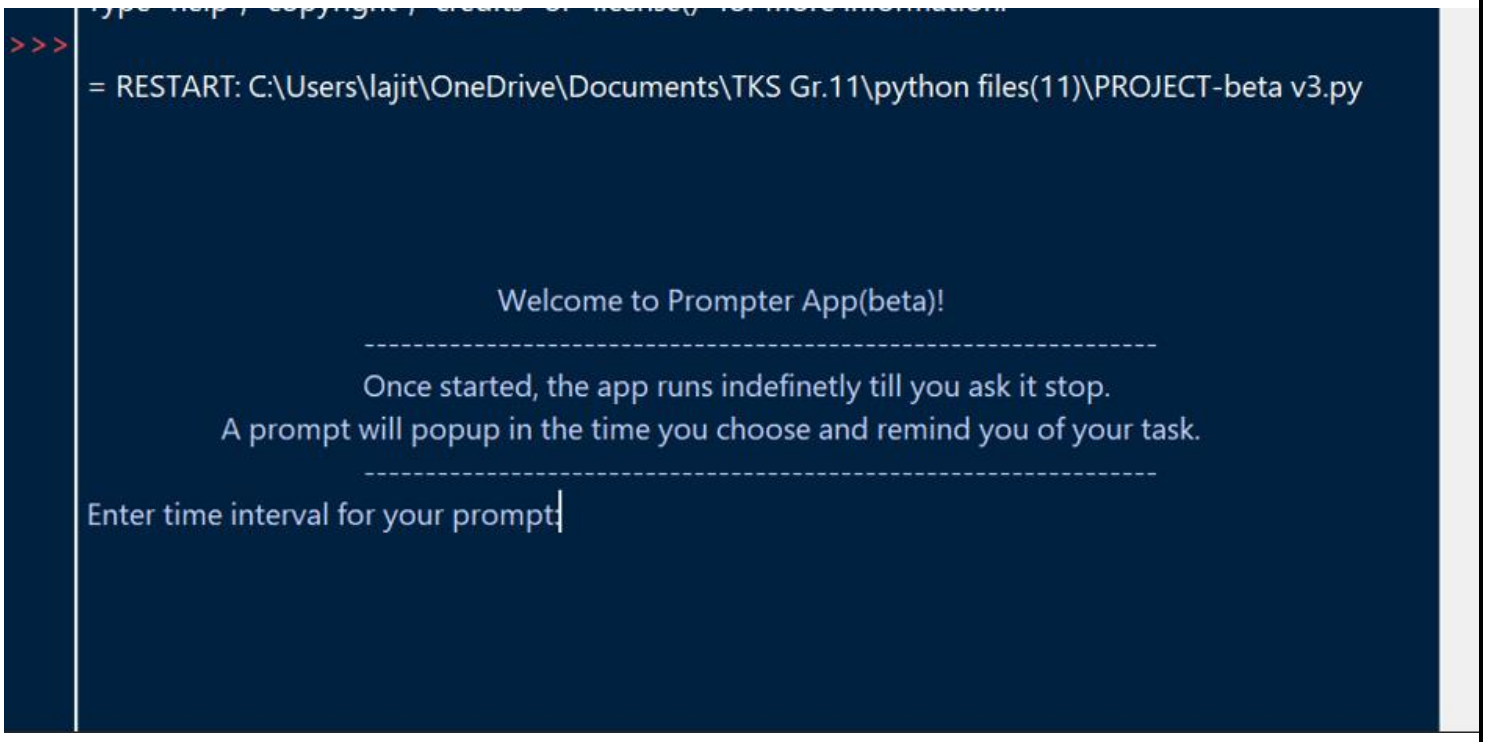
SOURCE CODE

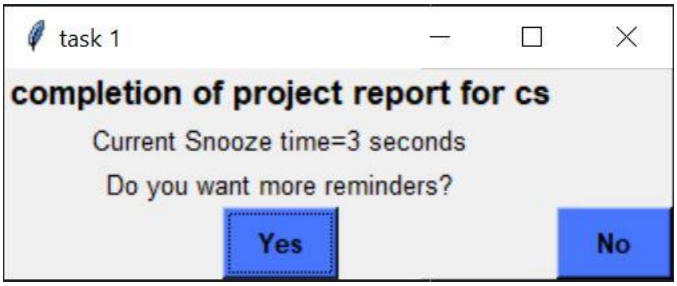
```
#importing modules

from time import sleep
from tkinter import *
import winsound

execution_count = 0

#defining functions

def action(win, more):
   global execution_count
   global root
   print('Answer', more)
   if more:
      win.destroy()
      sleep(snooze_time)
      # for now sleep is in seconds
      #for actual running:    snooze_hours=snooze_time*3600
      #                       sleep(sleep_hours)

      execution_count = execution_count + 1
      ReminderWindow(title, message)
      for i in range(0,2):
         winsound.Beep(300,500)
   else:
      win.destroy()
      root.destroy()

def ReminderWindow(title, message):
   global root
   print('\n\ntimes snoozed', execution_count)
   win = Tk()
   win.withdraw()
   win.update_idletasks()
```

```
    x = (win.winfo_screenwidth() - win.winfo_reqwidth()) / 2
    y = (win.winfo_screenheight() - win.winfo_reqheight()) / 2
    win.geometry("+%d+%d" % (x, y))
    win.foreground='sky blue'
    win.deiconify()
    win.title(title)

    message1=message
    message2='Current Snooze time={0} seconds'.format(snooze_time)
    message3 = 'Do you want more reminders?'
    Label(win, text=message1, font='arial 13 bold').grid(column=0,row=1)
    Label(win, text=message2, font='arial 10').grid(column=0,row=2)
    Label(win, text=message3, font='arial 10').grid(column=0,row=3)

    yes_btn = Button(win, text='Yes', font = 'arial 10 bold',pady = 5, command = lambda:
action(win, True),width = 6, bg = 'royal blue1', activebackground = 'sky blue')
    no_btn = Button(win, text='No', font = 'arial 10 bold',pady = 5,command = lambda:
action(win, False),width = 6, bg = 'royal blue1', activebackground = 'sky blue')
    yes_btn.grid(column=0,row=5)
    no_btn.grid(column=1, row=5)
    yes_btn.focus()
    win.lift()
    win.attributes('-topmost', True)




print('\n\n\n')
print('\t\t\t', 'Welcome to Prompter App(beta)!')
print('\t\t', '------------------------------------------------------------')
print('\t\t', 'Once started, the app runs indefinetly till you ask it stop.')
print('\tA prompt will popup in the time you choose and remind you of your task.')
print('\t\t', '------------------------------------------------------------')

#user-defined input
snooze_time = int(input('Enter time interval for your prompt:'))
title = input('Enter title for the prompt window: ')
message = input('Enter the task specifics/details: ')
```

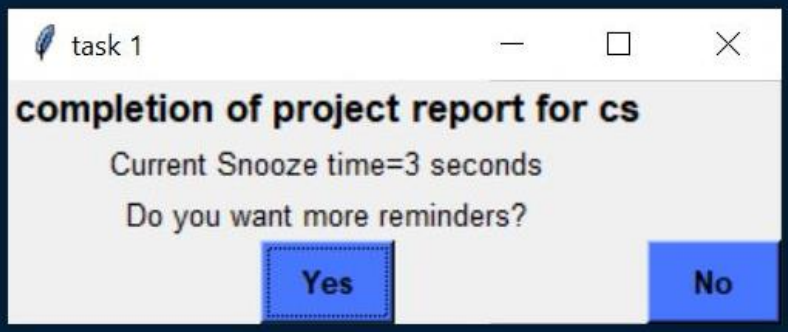
```
print('\n\nYou will get your first reminder in {0} seconds'.format(snooze_time))
print('\n\n')
print('App started...')

root = Tk()
root.withdraw()
execution_count = 1
sleep(snooze_time)
# for actual running:
# sleep(snooze_hours)
(ReminderWindow(title, message))
for i in range(0,2):
        winsound.Beep(400,500)
root.mainloop()


print('\n\nExiting Prompter App')
```

RESULTS (SCREENSHOTS)





```
                Welcome to Prompter App(beta)!
-----------------------------------------------------------------
        Once started, the app runs indefinetly till you ask it stop.
    A prompt will popup in the time you choose and remind you of your task.
-----------------------------------------------------------------
Enter time interval for your prompt:3
Enter title for the prompt window: task 1
Enter the task specifics/details: completion of project report for cs


You will get your first reminder in 3 seconds



App started...


times snoozed 1
Answer True


times snoozed 2
```



```
                    Welcome to Prompter App(beta)!
                    ------------------------------------------------------------
                    Once started, the app runs indefinetly till you ask it stop.
                A prompt will popup in the time you choose and remind you of your task.
                    ------------------------------------------------------------
Enter time interval for your prompt:3
Enter title for the prompt window: task 1
Enter the task specifics/details: completion of project report for cs


You will get your first reminder in 3 seconds



App started...


times snoozed 1
Answer True


times snoozed 2
Answer False


Exiting Prompter App
```

REFERENCES

- Computer Science with Python By: Preeti Arora
- https://www.youtube.com/watch?v=zF_DroDICaM
- https://github.com/
- https://stackoverflow.com/