DESKTOP NOTIFIER APPLICATION

As a project work for Course

PYTHON PROGRAMMING (INT 213)

Name : Rohit Roy
Registration Number : 12101227
Name : Sachin
Registration Number : 12100852
Program : CSE B.Tech Semester : Third
School : School of Computer Science and Engineering
Name of the University : Lovely Professional University

Date of submission :20th NOVEMBER 2021

Lovely Professional University
Jalandhar, Punjab, India.

DESKTOP NOTIFIER APPLICATION

20th NOVEMBER 2021

ABSTRACT:-

Nowadays People are too busy to remember things and keep track of their work,
So there's a need for a system that help them by reminding their work and keep them in track, so they can finish their work on time and manage their time
So we came up with a desktop notifier application that does the same for you.desktop notifier is a simple application which produces a notification message in the form of a pop-up message on desktop.

ACKNOWLEDGEMENT:-

I would like to thank my mentor - Prof. Sagar Pande for his advice and inputs on this project. Many thanks to my friends and seniors as well, who spent countless hours listening and providing feedback.

Table of Contents

INTRODUCTION:-

1.1 Context :-

This project has been done as part of my course for the CSE at Lovely Professional University . Supervised by Sagar Pande, I have three months to fulfill the requirements in order to succeed in the module.

1.2 Motivations :-

Being extremely interested in python specially in the making of GUI applications.the group project was a great occasion to give us the time to learn and confirm our interest for this field. We find out that we can use python in our day to day activities as a tool.That's why we decided to make a Desktop Notifier Application based on a problem that we face daily "forgetting things".

1.3 Idea:-

As a first experience, we wanted to make our project on a daily life problem and solve it.This project aims at building a desktop notifier system that will notify you after every specific period of time.The application runs on your system.The development of these apps makes the use of libraries and modules such as Plyer, Tkinter, Notification,Time,Image etc.

TEAM MEMBERS:-

TEAM LEADER:-

Rohit Roy:-

Contributions:-

1. Coding(joined)
2. Report work
3. GUI

Sachin:-

Contributions:-

1. Coding(joined)
2. Report work
3. GUI

LIBRARIES:-

Tkinter:-

Tkinter is the inbuilt python module that is used to create GUI applications. It is one of the most commonly used modules for creating GUI applications in Python as it is simple and easy to work with. You don't need to worry about the installation of the Tkinter module separately as it comes with Python already. It gives an object-oriented interface to the Tk GUI toolkit.

Plyer:-

Plyer is a Python library for accessing features of your hardware / platforms.

Time:-

Time This module provides various time-related functions. For related functionality, see also the datetime and calendar modules.

Although this module is always available, not all functions are available on all platforms. Most of the functions defined in this module call platform C library functions with the same name. It may sometimes be helpful to consult the platform documentation, because the semantics of these functions varies among platforms.

PIL :-

PIL Imaging Library (expansion of PIL) is the de facto image processing package for Python language. It incorporates lightweight image processing tools that aids in editing, creating and saving images. Support for Python Imaging Library got discontinued in 2011, but a project named pillow forked the original PIL project and added Python 3.x support to it. Pillow was announced as a replacement for PIL for future usage. Pillow supports a large number of image file formats including BMP, PNG, JPEG, and TIFF. The library encourages adding support for newer formats in the library by creating new file decoders.

MODULES:-

Image:-

Image module provides a class with the same name which is used to represent a PIL image. The module also provides a number of factory functions, including functions to load images from files, and to create new images.

ImageTk:-

ImageTk module contains support to create and modify Tkinter BitmapImage and PhotoImage objects from PIL images.

Messagebox:-

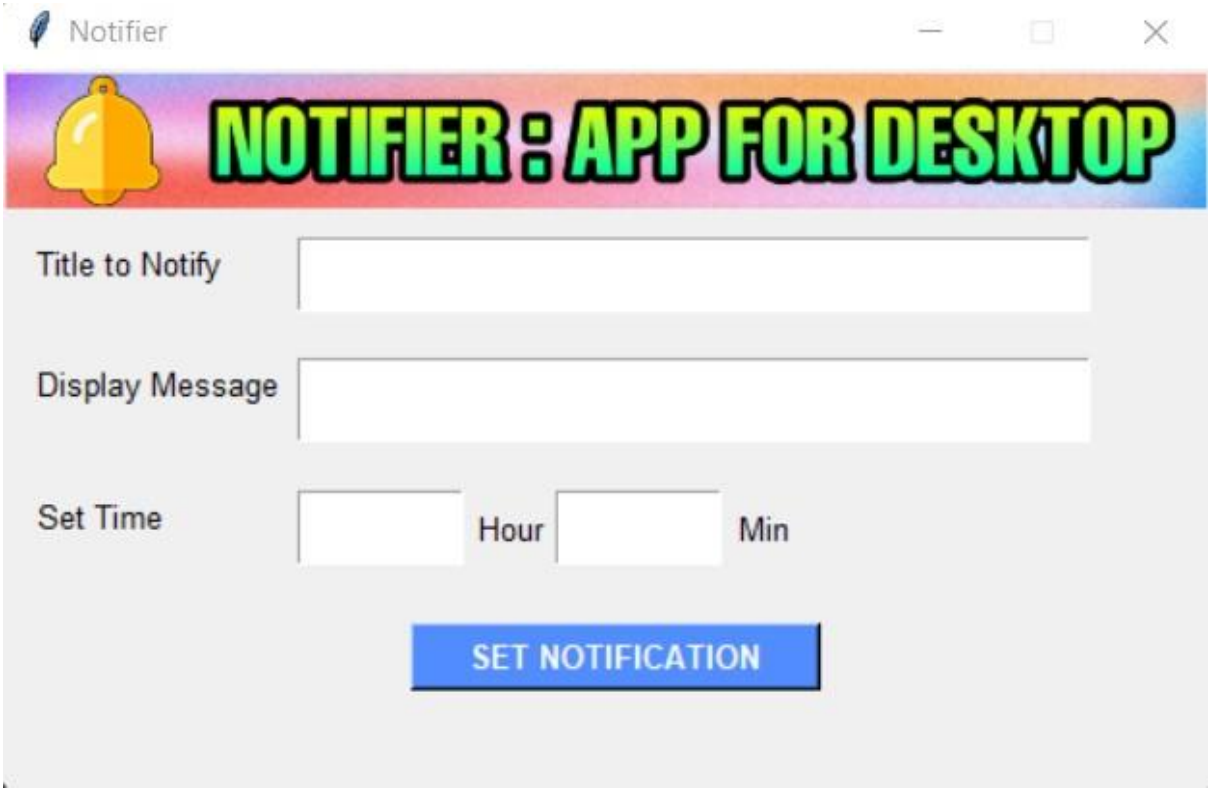
Tkinter.messagebox module provides a template base class as well as a variety of convenience methods for commonly used configurations. The message boxes are modal and will return a subset of (True, False, OK, None, Yes, No) based on the user's selection. Common message box styles and layout.

Notification:-

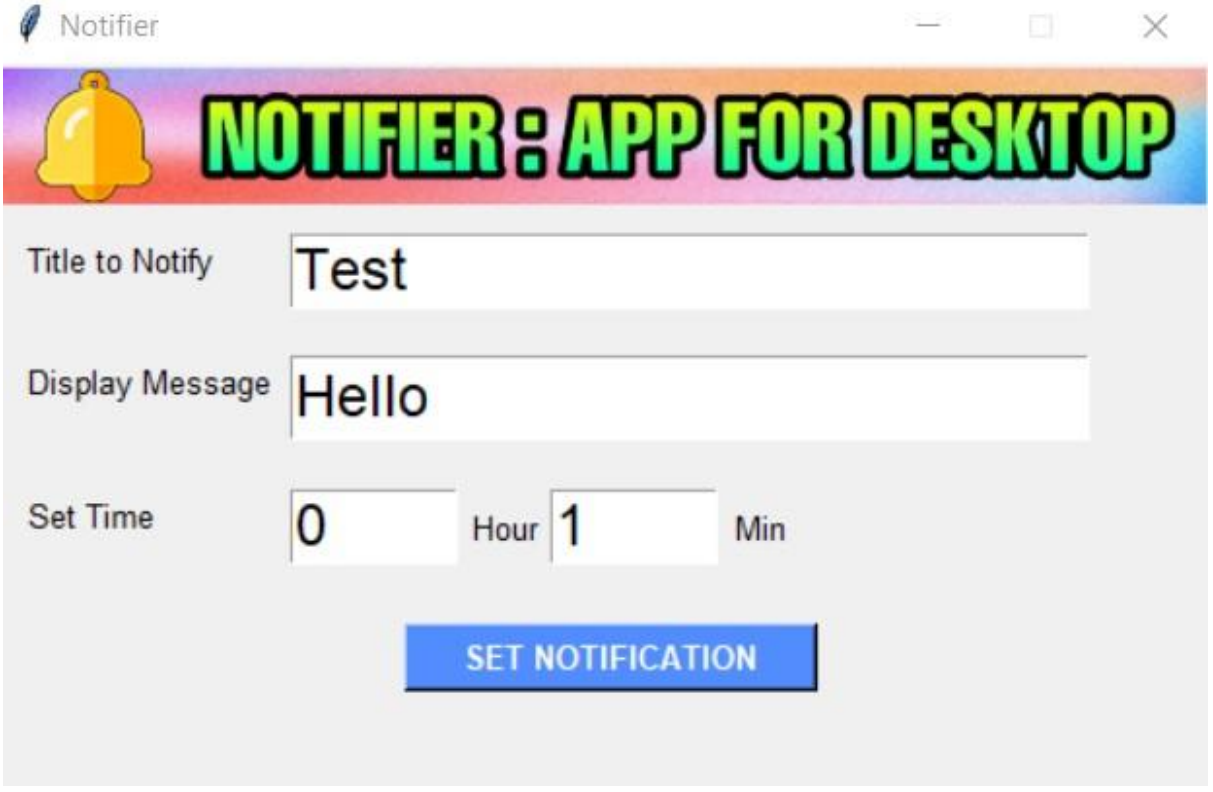
Notification module provides a Parameters:
title (str) – Title of the notification message (str) – Message of the notification
app_name (str) – Name of the app launching this notification
app_icon (str) – Icon to be displayed along with the message
timeout (int) – time to display the message for, defaults to 10
ticker (str) – text to display on status bar as the notification arrives
toast (bool) – simple Android message instead of full notification

SCREENSHOTS:-

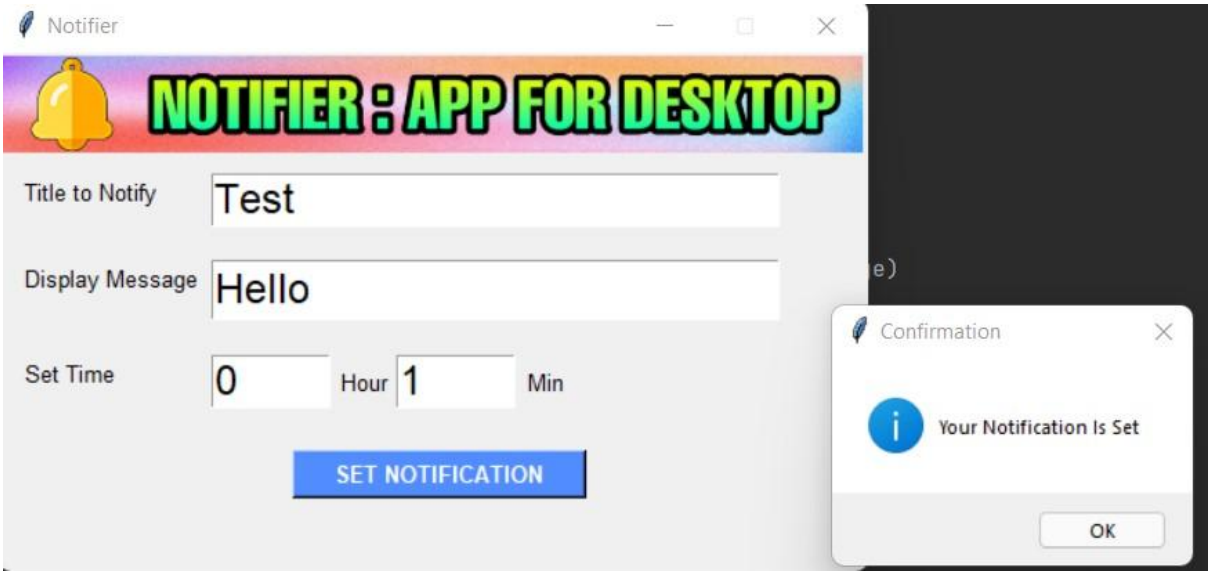
1.Home Page :-



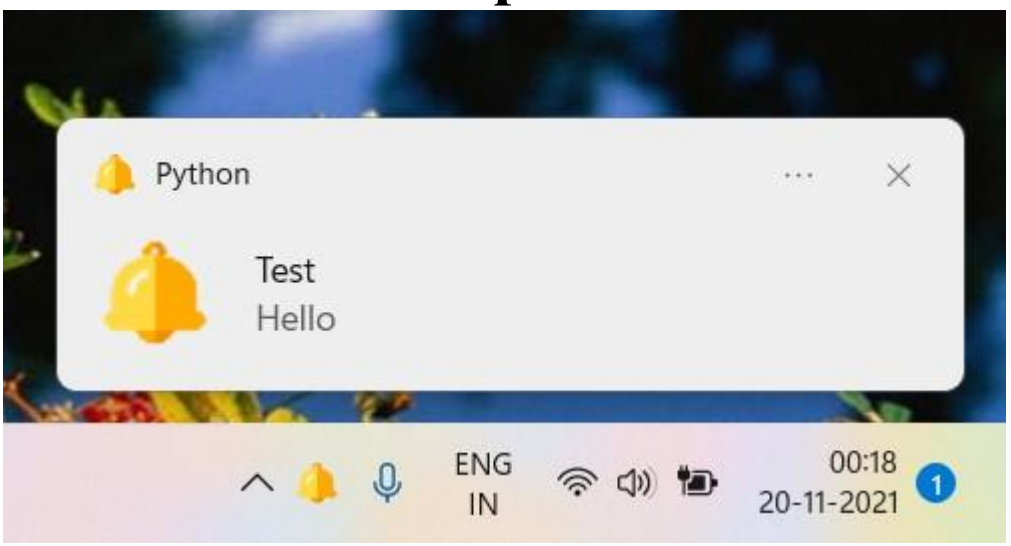
2.Input Data Into It :-



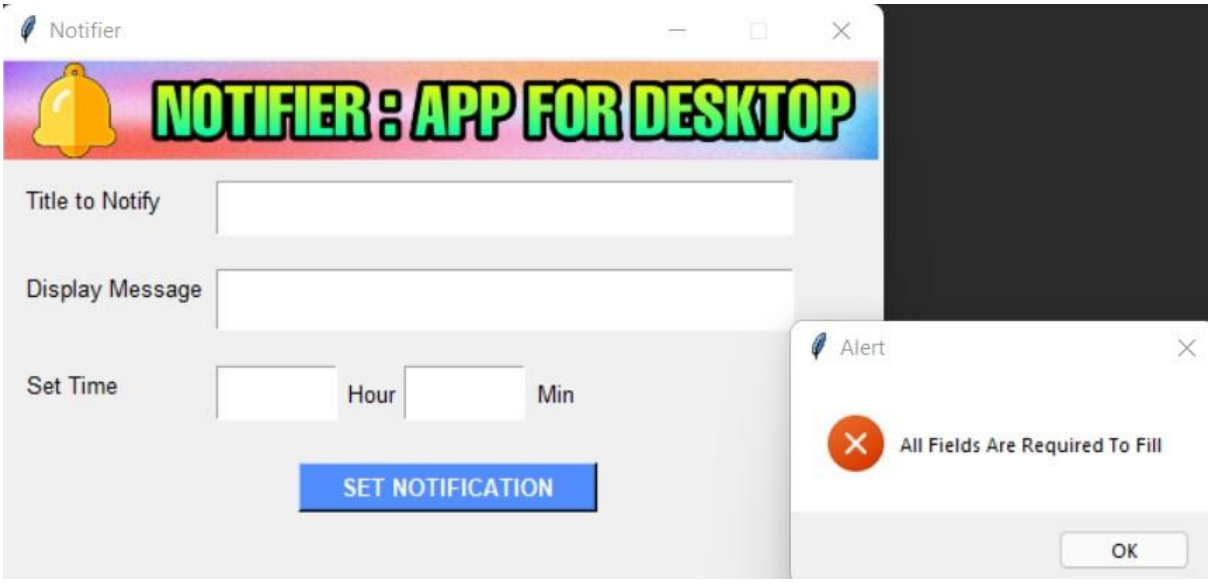
3.Confirmation Message :-



4.Notification Output :-



5.Without Inputting Any Data :-



6.Codes :-

main.py

```
from tkinter import *
from plyer import notification
from tkinter import messagebox
import time
from PIL import Image, ImageTk


t = Tk()
t.title('Notifier')
image = Image.open("bg.png")
t.geometry("503x300")
tkimage = ImageTk.PhotoImage(image)


def get_details():
    get_title = title.get()
    get_message = message.get()
    get_time = time1.get()
    get_time1 = time2.get()

    if get_title == "" or get_message == "" or get_time == "":
        messagebox.showerror("Alert", "All Fields Are Required To Fill")
    else:
        int_time = int(float(get_time))
        min_to_sec = int_time * 60
        int_time1 = int(float(get_time1))
        min_to_hours = int_time1 * 60 * 60
        messagebox.showinfo("Confirmation", "Your Notification Is Set")
        t.destroy()
        time.sleep(min_to_sec)
```

Python Console | Terminal | Event Log

68:28 CRLF UTF-8 4 spaces Python 3.9 (DesktopNotifierApp)

main.py

```
        time.sleep(min_to_hours)

        notification.notify(title=get_title,
                            message=get_message,
                            app_name="Notifier",
                            app_icon="logo.ico",
                            toast=True,
                            timeout=60)


image_label = Label(t, image=tkimage).grid()

title_label = Label(t, text="Title to Notify", font=("poppins", 10))
title_label.place(x=12, y=70)

title = Entry(t, width="25", font=("poppins", 18))
title.place(x=123, y=70)

message_label = Label(t, text="Display Message", font=("poppins", 10))
message_label.place(x=12, y=120)

message = Entry(t, width="25", font=("poppins", 18))
message.place(x=123, height=35, y=120)

time_label = Label(t, text="Set Time", font=("poppins", 10))
time_label.place(x=12, y=175)

time1 = Entry(t, width="5", font=("poppins", 18))
time1.place(x=230, y=175)
```

Python Console | Terminal | Event Log

68:28 CRLF UTF-8 4 spaces Python 3.9 (DesktopNotifierApp)

main.py

```
message = Entry(t, width="25", font=("poppins", 18))
message.place(x=123, height=35, y=120)

time_label = Label(t, text="Set Time", font=("poppins", 10))
time_label.place(x=12, y=175)

time1 = Entry(t, width="5", font=("poppins", 18))
time1.place(x=230, y=175)

time_min_label = Label(t, text="Min", font=("poppins", 10))
time_min_label.place(x=303, y=180)

time2 = Entry(t, width="5", font=("poppins", 18))
time2.place(x=123, y=175)

time_hour_label = Label(t, text="Hour", font=("poppins", 10))
time_hour_label.place(x=195, y=180)

button = Button(t, text="SET NOTIFICATION", font=("poppins", 10, "bold"), fg="#ffffff", bg="#528DFF", width=20,
                relief="raised", command=get_details)
button.place(x=170, y=230)
t.resizable(0, 0)
t.mainloop()
```

Python Console | Terminal | Event Log

68:28 CRLF UTF-8 4 spaces Python 3.9 (DesktopNotifierApp)

Conclusions:-

It is our team's hope that this document will be of huge help with understanding of our little project and our approach to solve the problem of forgetting things .

REFERENCES:-

To conduct this project the following tools have been used :

- Pycharm(IDE) https://www.jetbrains.com/pycharm/
- Tkinter(library) https://docs.python.org/3/library/tkinter.html
- Plyer(library) https://plyer.readthedocs.io/en/latest/
- PIL(library)https://python-pillow.org/

To conduct this project the following websites have been used:

- GeeksforGeeks https://www.geeksforgeeks.org/
- YouTube Channel https://www.youtube.com/c/SDPGuruji/