**Countdown Timer and Notification Application**

**Overview**

This application is a simple countdown timer with desktop notifications. It allows users to set a timer by specifying hours, minutes, and seconds. Once the timer reaches zero, a desktop notification is displayed, and a message box pops up to alert the user.

**Features**

* Set a countdown timer using hours, minutes, and seconds.
* Displays the remaining time in the application window.
* Shows a desktop notification when the timer ends.
* Provides a message box alert when the timer completes.

**Requirements**

* **Python 3.x**: Ensure Python 3.x is installed.
* **Plyer**: For cross-platform notifications.
* **Tkinter**: For the graphical user interface.

**Installation**

1. **Install Plyer**:

pip install plyer

**Tkinter**: Tkinter comes bundled with Python's standard library, so no additional installation is usually required.

**Code Explanation**

**Importing Modules**

from plyer import notification

from tkinter import messagebox

from tkinter import \*

import time

**plyer**: Used for sending desktop notifications.

**tkinter**: Used for creating the graphical user interface.

**time**: Used for creating delays in the countdown.

**Creating the GUI**

# Initialize the main window

window = Tk()

window.geometry("300x200")

window.title("Countdown timer and notification")

window.config(bg='lightblue')

* Initializes the main window with a specified size and title.
* Sets the background color to lightblue.

**Entry Placeholder Removal**

def h\_click(event):

hour\_entry.delete(0, 'end')

def m\_click(event):

min\_entry.delete(0, 'end')

def s\_click(event):

sec\_entry.delete(0, 'end')

* **h\_click**: Clears the hour entry field when clicked.
* **m\_click**: Clears the minute entry field when clicked.
* **s\_click**: Clears the second entry field when clicked.

**Timer Function**

def timer():

try:

timer\_time = int(hour\_entry.get())\*3600 + int(min\_entry.get())\*60 + int(sec\_entry.get())

except:

messagebox.showerror(message="Enter Valid Time")

if timer\_time > 0:

hour = 0

min = 0

sec = 0

while timer\_time >= 0:

min, sec = divmod(timer\_time, 60)

if min > 60:

hour, min = divmod(min, 60)

hours.set(hour)

mins.set(min)

secs.set(sec)

time.sleep(1)

window.update()

timer\_time -= 1

notification.notify(

title="TIMER ALERT",

message="Hey kodi!\nDid you do what you wanted to achieve? \nIf not, try again with a new timer",

timeout=30,

)

messagebox.showinfo(message="Timer Complete!")

**timer**: Calculates the total timer duration in seconds. Updates the displayed time every second. Sends a notification and shows a message box when the timer finishes.

**Creating GUI Elements**

title\_label\_1 = Label(window, text="Timer with notification", font=("Gayathri", 12), bg='lightblue')

title\_label\_1.pack()

title\_label\_2 = Label(window, text="Put 0 in fields not of use", font=("Gayathri", 10), bg='lightblue')

title\_label\_2.pack()

hours = IntVar()

mins = IntVar()

secs = IntVar()

hour\_entry = Entry(window, width=3, textvariable=hours, font=("Ubuntu Mono", 18), bg='white')

min\_entry = Entry(window, width=3, textvariable=mins, font=("Ubuntu Mono", 18), bg='white')

sec\_entry = Entry(window, width=3, textvariable=secs, font=("Ubuntu Mono", 18), bg='white')

hour\_entry.insert(0, 00)

min\_entry.insert(0, 00)

sec\_entry.insert(0, 00)

hour\_entry.place(x=80, y=40)

min\_entry.place(x=130, y=40)

sec\_entry.place(x=180, y=40)

hour\_entry.bind("<1>", h\_click)

min\_entry.bind("<1>", m\_click)

sec\_entry.bind("<1>", s\_click)

button = Button(window, text='Start Timer', bg='cyan', command=timer)

button.pack(pady=40)

* **Labels**: Display titles and instructions.
* **Entry Widgets**: Allow user input for hours, minutes, and seconds. Background color set to white.
* **Button**: Starts the timer when clicked.

**Main Loop**

window.mainloop()

**Starts the Tkinter event loop, which keeps the application running and responsive to user interactions.**