Technological Institute of the Philippines
Computer Engineering Department
Quezon city Campus

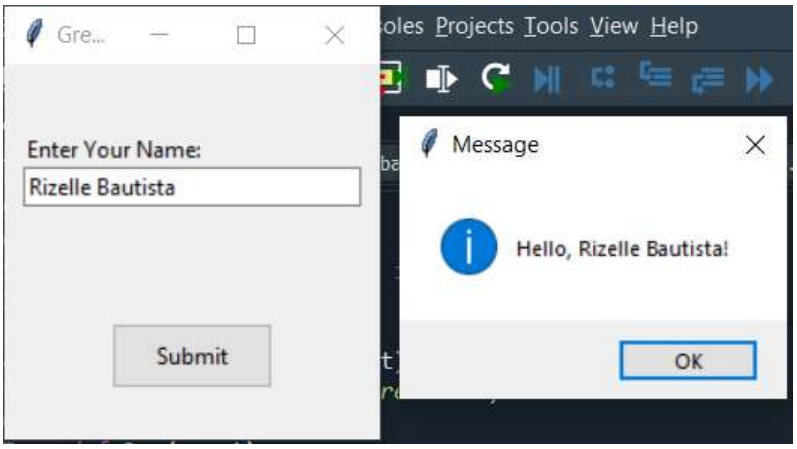
Hands-On-Activity 9.1 – Introduction to Tkinter	
Name	Bautista, Rizelle B.
Course and Section	CPE 009 - Object Oriented Programming - CPE12S3
Instructor	Dr. Jonathan V. Taylar

TASK:

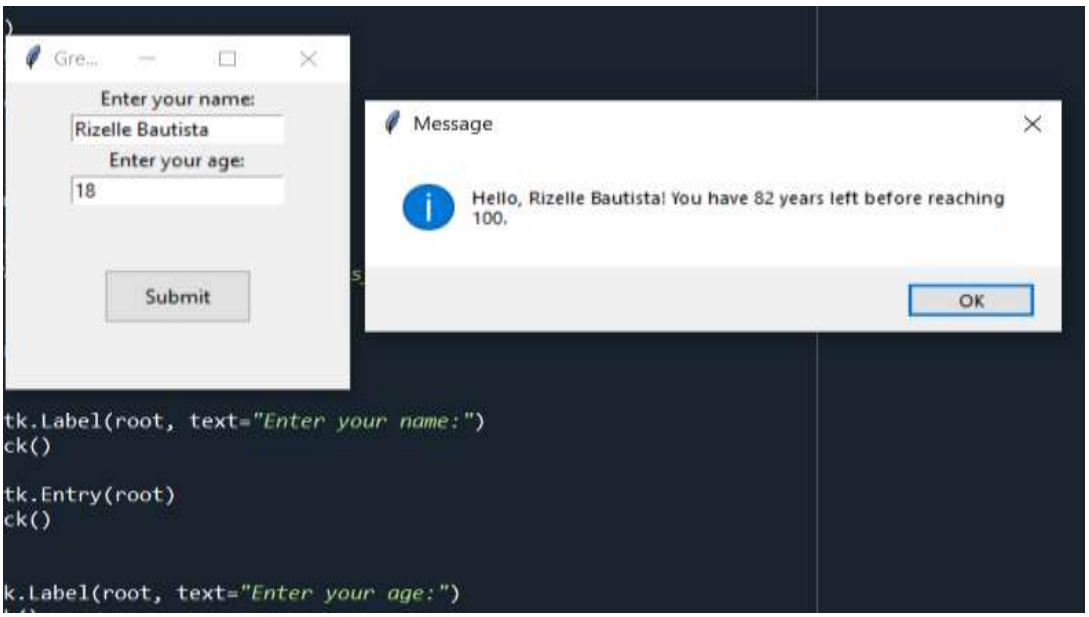
1. Create a GUI application that allows entering a name and outputs the following as shown when pressing the return key on your keyboard:

```
import tkinter as tk
from tkinter import *
from tkinter import ttk
from tkinter.messagebox import showinfo


def return_pressed(event):
    print('Return key pressed.')

def log(event):
    print(event)

root = tk.Tk()
root.title('Greeting Application')
root.geometry("200x200")
root.resizable(True, True)


name = tk.StringVar()


def Greetings():
    msg = f'Hello, {name.get()}!'
    showinfo(
        title = 'Message',
        message = msg)

names = ttk.Frame(root)
names.pack(padx =10, pady = 10, fill = 'x', expand = True)

name_label = ttk.Label(names, text ='Enter Your Name: ')
name_label.pack(fill = 'x', expand = True)
name_entry = ttk.Entry(names, textvariable = name)
name_entry.pack(fill = 'x', expand = True)
name_entry.focus()
```

```
myButton = ttk.Button(root, text = "Submit", command= Greetings)
myButton.bind('<Return>', return_pressed)
myButton.bind('<Return>', log, add = '+')


myButton.focus()
myButton.pack(ipadx = 5, ipady = 5, expand = True)


root.mainloop()
```



2. Create an application that allows entering a name and an age, when the Submit button is clicked the output image on the right is shown that computes how many years are left for the person before reaching 100.



```python
import tkinter as tk
from tkinter import ttk
from tkinter.messagebox import showinfo


def return_pressed(event):
    print('Return key pressed.')

def log(event):
    print(event)

root = tk.Tk()
root.title('Greeting Application')
root.geometry("200x200")
root.resizable(True, True)

def Greetings():

    name = name_entry.get()
    age = int(age_entry.get())
    years_left = 100 - age
    msg = f"Hello, {name}! You have {years_left} years left before reaching 100."
    showinfo(
        title = 'Message',
        message = msg)
```

```python
name_label = tk.Label(root, text="Enter your name:")
name_label.pack()

name_entry = tk.Entry(root)
name_entry.pack()


age_label = tk.Label(root, text="Enter your age:")
age_label.pack()

age_entry = tk.Entry(root)
age_entry.pack()



myButton = ttk.Button(root, text = "Submit", command= Greetings)
myButton.bind('<Return>', return_pressed)
myButton.bind('<Return>', log, add = '+')


myButton.focus()
myButton.pack(ipadx = 5, ipady = 5, expand = True)


root.mainloop()
```

Supplementary Activity:

- **Log-In Python Script:**

```
import tkinter as tk
from tkinter import ttk
from tkinter.messagebox import showinfo

root = tk.Tk()
root.title('CAINTA EMERGENCY HOTLINE')
root.iconbitmap( 'icon.ico' )

# Set the window size and make it non-resizable
window_width = 1300
window_height = 700
root.resizable(False, False)

# setting the position of the window to the center
screen_width = root.winfo_screenwidth()
screen_height = root.winfo_screenheight()
center_x = int(screen_width / 2 - window_width / 2)
center_y = int(screen_height / 2 - window_height / 2)
root.geometry(f'{window_width}x{window_height}+{center_x}+{center_y}')

# frame for bg
frame = tk.Frame(root, bg="white", width=1300, height=700)
frame.pack(expand=False)

# picture background
bg_img = tk.PhotoImage(file="asdfas.png")
bg_label = tk.Label(frame, image=bg_img)
bg_label.place(x=0, y=0, relwidth=1, relheight=1)
```

Using the Tkinter library, this code creates a window for a CAINTA Emergency Hotline. It makes the window non-resizable and sets the size to 1300x700. The code also places the window in the center of the screen and uses a white backdrop. It also decorates the window with a background image.

```
# register label
register_label = tk.Label(frame, text="Login", font=("Arial", 15))
register_label.place(relx=0.14, rely=0.25, anchor="center")

# username label and input
username_label = tk.Label(frame, text="Username", font=("Helvetica", 9))
username_label.place(relx=0.1, rely=0.29, anchor="e")
username_entry = tk.Entry(frame, font=("Helvetica", 12))
username_entry.place(relx=0.05, rely=0.32, anchor="w")

# password label and input
password_label = tk.Label(frame, text="Password", font=("Helvetica", 9))
password_label.place(relx=0.1, rely=0.35, anchor="e")
password_entry = tk.Entry(frame, show="*", font=("Helvetica", 12))
password_entry.place(relx=0.05, rely=0.38, anchor="w")


# button/picture
def register_clicked():
    showinfo(title='info', message='registered')
    reg_icon = tk.PhotoImage(file='button.png')
    reg_button = ttk.Button(root, image=reg_icon, text=' ',
                            command=register_clicked)
    reg_button.pack(ipadx=5, ipady=5, expand=False)

root.mainloop()
```

In this part of the code, we develop here the title, icon and the size of the login window and also the background picture of the login window, for now we are still working on the registration part of the program.

- **Home Page Code**

```
import tkinter as tk


root = tk.Tk()
root.geometry('1080x600')
root.title('Home Page')


#Display the frame of the home page
def home_page():
    home_frame = tk.Frame(main_frame)

    lb = tk.Label(home_frame, text = 'Info abt hotlines',font=('Bold', 4    0))
    lb.pack()
    home_frame.pack(pady=20)

#Display the frame of the profile page
def profile_page():
    profile_frame = tk.Frame(main_frame)

    lb = tk.Label(profile_frame, text = 'User Profile',font=('Bold', 40))
    lb.pack()
    profile_frame.pack(pady=20)

#Display the frame of the settings page
def settings_page():
    settings_frame = tk.Frame(main_frame)

    lb = tk.Label(settings_frame, text = 'Settings',font=('Bold', 40))
    lb.pack()
    settings_frame.pack(pady=20)

#Display the frame of the about page
def about_page():
    about_frame = tk.Frame(main_frame)

    lb = tk.Label(about_frame, text = 'App description/ may discard it later',font=('Bold', 40))
    lb.pack()
    about_frame.pack(pady=20)
```

We developed a GUI for the homepage of our application using this Python script. The main frame and options frame's size, text, and background color have all been set for the time being. For now, we've been working on the menu bar and frame buttons, as well as how to change the windows frame. As for the other parts of the GUI, it is still working in progress as we want to deliver our very best in bringing the best service in our application.

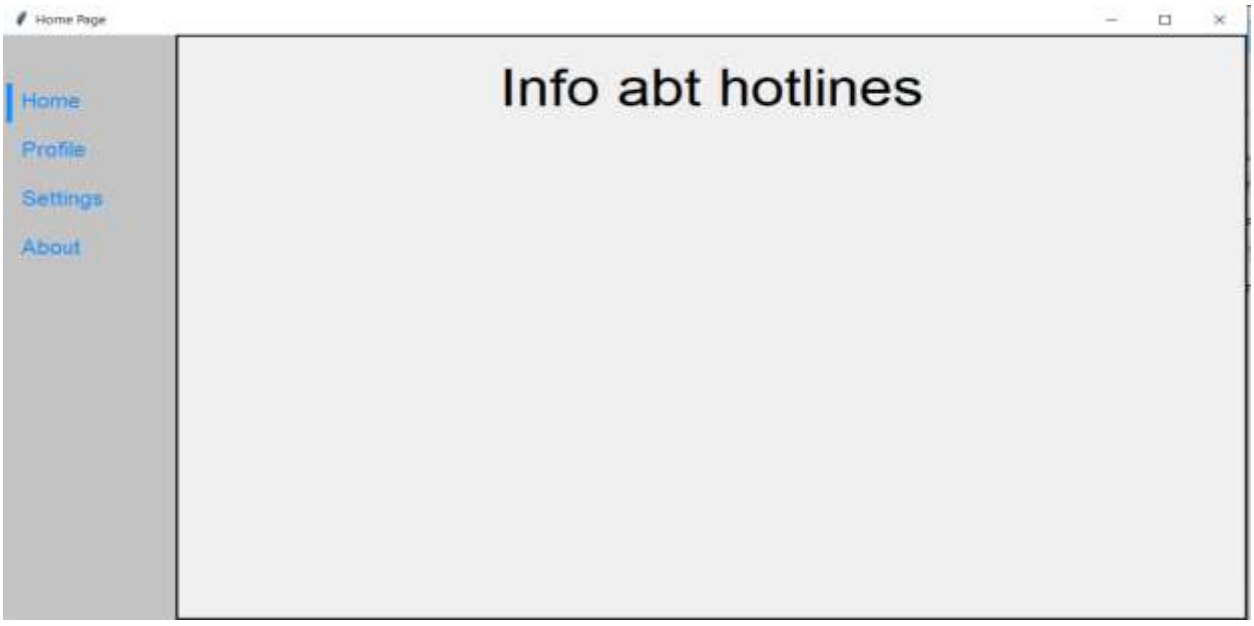
```
#Set the position, size, and background color of the settings button
settings_btn = tk.Button(options_frame, text = 'Settings', font = ('Bold', 15),
                  fg = '#158aff', bd = 0, bg = '#c3c3c3',
                  command = lambda: indicate(settings_indicate, settings_page))

settings_btn.place(x=10, y =150)
settings_indicate = tk.Label(options_frame, text = '', bg = '#c3c3c3')
settings_indicate.place(x=3,y = 150, width = 5, height=40)


#Set the position, size, and background color of the about button
about_btn = tk.Button(options_frame, text = 'About', font = ('Bold', 15),
                  fg = '#158aff', bd = 0, bg = '#c3c3c3',
                  command = lambda: indicate(about_indicate, about_page))


about_btn.place(x=10, y =200)
about_indicate = tk.Label(options_frame, text = '', bg = '#c3c3c3')
about_indicate.place(x=3,y = 200, width = 5, height=40)

#Set the height and width of the options frame
options_frame.pack(side=tk.LEFT)
options_frame.pack_propagate(False)
options_frame.configure(width=150, height=600)

#Set the background color of the main frame
main_frame = tk.Frame(root, highlightbackground='black',
                      highlightthickness = 2)

#Set the  height and width of the main frame
main_frame.pack(side=tk.LEFT)
main_frame.pack_propagate(False)
main_frame.configure(height=600,width=1080)

root.mainloop()
```

The user interface is divided into two frames: the choices frame and the mainframe. The choices frame has buttons that allow the user to choose between various settings and pages, while the main frame shows the contents of the selected page. The code determines the location, size, and color of the settings and buttons, as well as the height and width of the options frame and main frame. The code also assigns a command to each button to indicate which page is selected and then displays the page's contents in the mainframe. Finally, the code concludes with the root.mainloop() instruction, which runs the GUI.

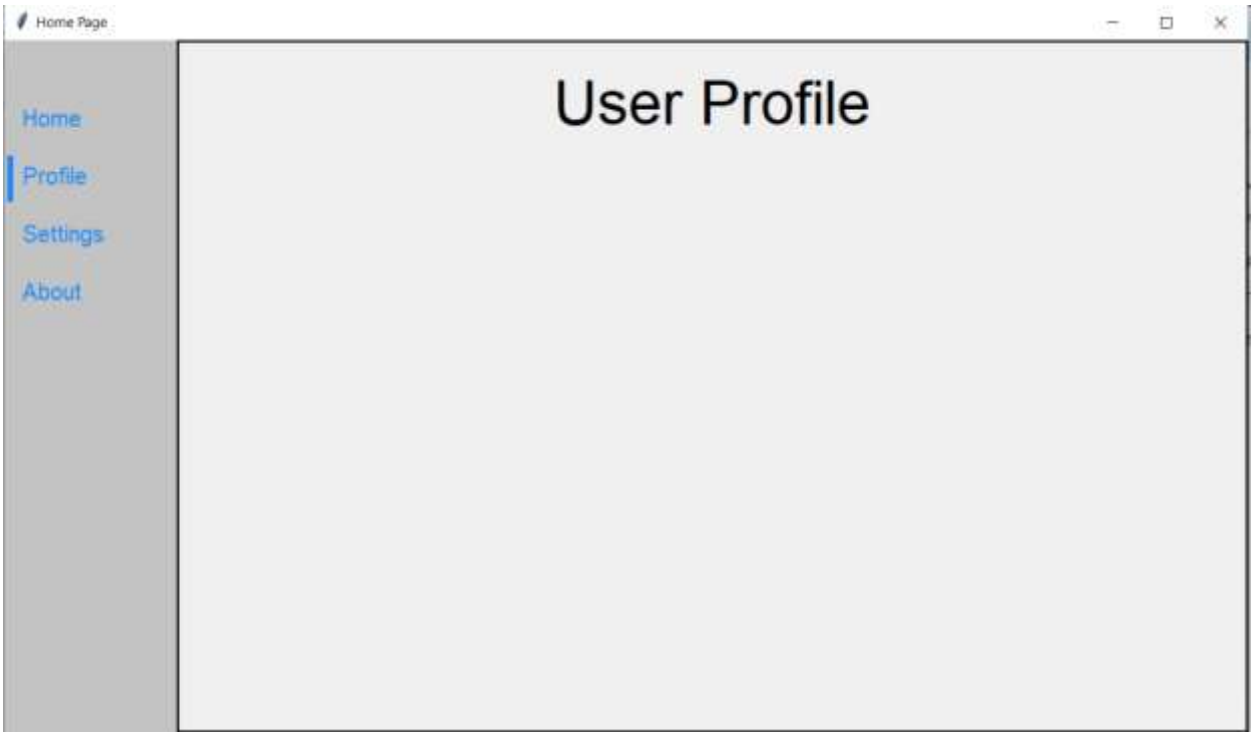
- **Log-In Output**

- **Login page for the user and administrator. Registration for users coming soon.**

- **Home Page Output**

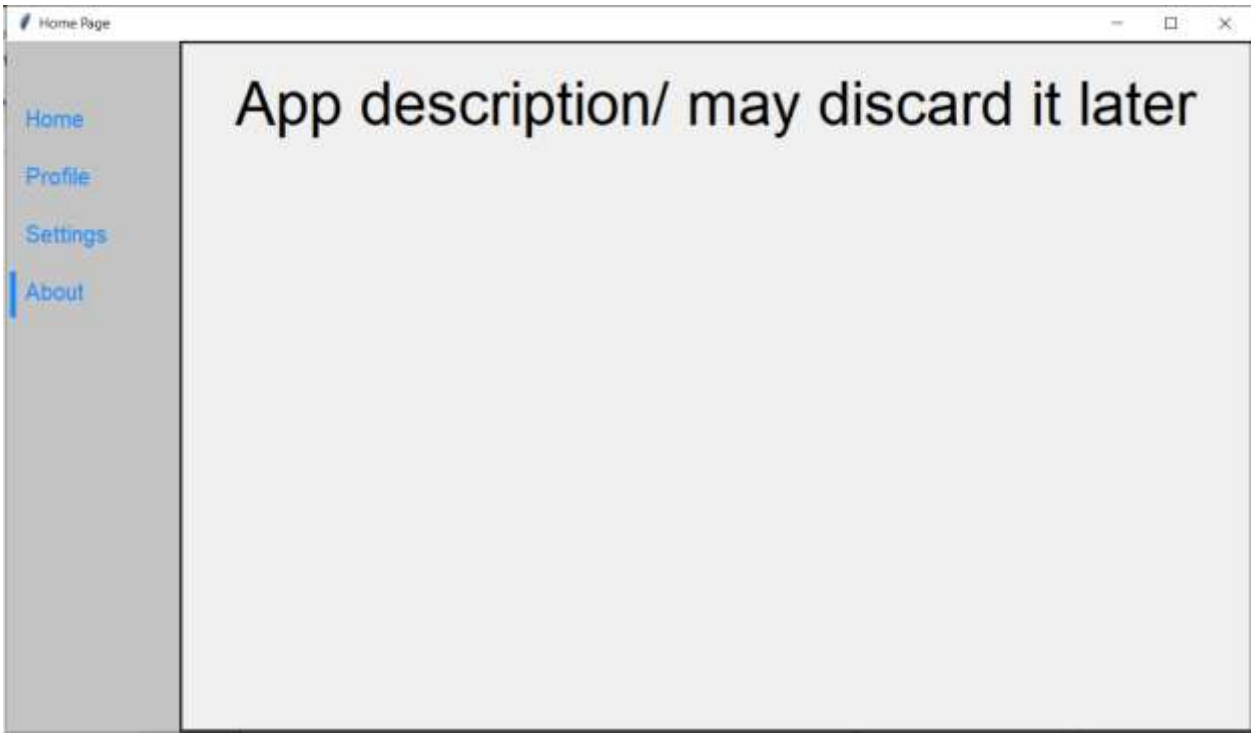


- **Compilation of all emergency hotlines in Cainta.**



- **Page for user profile containing the user's information.**

- **Settings**



- **Information about the creators and the app itself**

Conclusion:

I learned how to use the Tkinter GUI and its features such as event and command binding through this activity. By incorporating labels, buttons, and graphics, our codes come to life. It adds a GUI, which makes our code look nicer. Using labels is like labeling a widget. It's comparable to giving something a name in the user interface that conveys its purpose. Correct labeling is essential to prevent user confusion when they interact with it.

I also picked up some Tkinter event binding skills. I believe that assigning keyboard buttons to interact with the GUI is what is meant by event binding. When inputting text or using an interface, the return key function is the one that is most frequently utilized. It is challenging since you must be aware of the keyword. Overall, this exercise gives me the chance to explore Tkinter further and see what I can accomplish with the code we produced for our final project.