Smart Home GUI - Salman AlMaskati

In [ ]:

```
from phue import Bridge
from tkinter import *
import tkinter as tk
from tkinter import ttk
from tkinter import simpledialog
from tkinter import messagebox
import random
import time
from PIL import Image, ImageTk
from tkinter import colorchooser


#link bridge
b = Bridge('192.168.100.68')
b.connect()
b.get_api()


#Function defs

#LIVING ROOM FUNCTIONS
def livingroom_on():
    b.set_group(3,'on',True)

def livingroom_off():
    b.set_group(3,'on',False)


def livingroom_dimmer():
    #new window
    new_window = Tk()
    new_window.title("Living Room Dimmer Window")
    new_window.geometry('200x200')
    scale = Scale(new_window, from_=0, to=254)
    scale.grid(row=1,column=1)
    current_bright=b.get_group(3, 'bri')
    scale.set(current_bright)


    def get_scale():
        brightness=scale.get()
        brightness=int(brightness)
        print(brightness)
        scale_get.grid(row=2,column=1)
        b.set_group(3,'on',True) #turn on lights
        b.set_light(5, 'bri', brightness) #adjust brightness
        b.set_light(4, 'bri', brightness)


    scale_get = tk.Button(new_window, text="Confirm", command=get_scale)
    scale_get.grid(row=2,column=1)

#SALMAN BEDROOM FUNCTIONS
def salmanroom_on():
```

```
    b.set_group(1,'on',True)

def salmanroom_off():
    b.set_group(1,'on',False)

def salman_dimmer():
    #new window
    new_window = Tk()
    new_window.title("Salman Bedroom Dimmer Window")
    new_window.geometry('200x200')
    scale = Scale(new_window, from_=0, to=254)
    scale.grid(row=1,column=1)
    current_bright=b.get_group(1, 'bri')
    scale.set(current_bright)


    def get_scale():
        brightness=scale.get()
        brightness=int(brightness)
        print(brightness)
        scale_get.grid(row=2,column=1)
        b.set_group(1,'on',True) #turn on lights
        b.set_light(2, 'bri', brightness) #adjust brightness
        b.set_light(4, 'bri', brightness)
        b.set_light(1, 'bri', brightness)
        b.set_light(7, 'bri', brightness)

    scale_get = tk.Button(new_window, text="Confirm", command=get_scale)
    scale_get.grid(row=2,column=1)




#AHMED BEDROOM FUNCTION
def ahmedroom_on():
    b.set_group(4,'on',True)

def ahmedroom_off():
    b.set_group(4,'on',False)

def ahmed_dimmer():
    #new window
    new_window = Tk()
    new_window.title("Ahmed Bedroom Dimmer Window")
    new_window.geometry('200x200')
    scale = Scale(new_window, from_=0, to=254)
    scale.grid(row=1,column=1)
    current_bright=b.get_group(4, 'bri')
    scale.set(current_bright)

    def get_scale():
        brightness=scale.get()
        brightness=int(brightness)
        print(brightness)
        scale_get.grid(row=2,column=1)
        b.set_group(4,'on',True) #turn on lights
        b.set_light(9, 'bri', brightness) #adjust brightness
```

```
    scale_get = tk.Button(new_window, text="Confirm", command=get_scale)
    scale_get.grid(row=2,column=1)


#ALL LIGHTS FUNCTION
def all_on():
    b.set_group(1,'on',True)
    b.set_group(3,'on',True)
    b.set_group(4,'on',True)

def all_off():
    b.set_group(1,'on',False)
    b.set_group(3,'on',False)
    b.set_group(4,'on',False)

def all_dimmer():
    #new window
    new_window = Tk()
    new_window.title("All Lights Dimmer Window")
    new_window.geometry('200x200')
    scale = Scale(new_window, from_=0, to=254)
    scale.grid(row=1,column=1)
    current_bright=b.get_group(1, 'bri')
    scale.set(current_bright)


    def get_scale():
        brightness=scale.get()
        brightness=int(brightness)
        print(brightness)
        scale_get.grid(row=2,column=1)
        b.set_group(1,'on',True) #turn on lights
        b.set_group(3,'on',True) #turn on lights
        b.set_group(4,'on',True) #turn on lights

        b.set_light(2, 'bri', brightness) #adjust brighntess
        b.set_light(4, 'bri', brightness)
        b.set_light(1, 'bri', brightness)
        b.set_light(7, 'bri', brightness)

        b.set_light(5, 'bri', brightness)
        b.set_light(6, 'bri', brightness)

        b.set_light(9, 'bri', brightness)

    scale_get = tk.Button(new_window, text="Confirm", command=get_scale)
    scale_get.grid(row=2,column=1)



                        #COLORS FUNCTION
def color_window():
    new_window = Tk()
    new_window.title("Color Window")

    label = ttk.Label(new_window, text="Select a Room:")
    label.pack()

    rooms = {'Salman Bedroom': ['1', '2', '4', '7'],
            'Living Room': ['5', '6'],
```

```
        'Ahmed Bedroom': ['9']
        }
combo = ttk.Combobox(new_window, values=list(rooms.keys()))
combo.pack()
def get_selected_value():
    global selected_value
    selected_value = combo.get()
    print("Selected Value:", selected_value)
    print(b.get_group(selected_value))
    global selected_value_gid
    selected_value_gid= b.get_group_id_by_name(selected_value)
    print(selected_value_gid)

button = tk.Button(new_window, text="Confrim", command=get_selected_value)
button.pack()




#RGB CONVERSTION
def rgb_to_xy(red, green, blue):

    # gamma correction
    red = pow((red + 0.055) / (1.0 + 0.055), 2.4) if red > 0.04045 else (re
    green = pow((green + 0.055) / (1.0 + 0.055), 2.4) if green > 0.04045 el
    blue =  pow((blue + 0.055) / (1.0 + 0.055), 2.4) if blue > 0.04045 else

    # convert rgb to xyz
    x = red * 0.649926 + green * 0.103455 + blue * 0.197109
    y = red * 0.234327 + green * 0.743075 + blue * 0.022598
    z = green * 0.053077 + blue * 1.035763

    # convert xyz to xy
    x = x / (x + y + z)
    y = y / (x + y + z)

    # TODO check color gamut if known

    return [x, y]

def choose_color():

    color_code = colorchooser.askcolor(title ="Choose color")
    rgb_color = color_code[0]
    #print(rgb_color[0])
    #print(rgb_color[1])
    #print(rgb_color[2])
    xy_color=rgb_to_xy(rgb_color[0],rgb_color[1],rgb_color[2])
    #print("XY Color:", xy_color)
    print(selected_value_gid)
    b.set_group(selected_value_gid,'xy',xy_color)

colors = ttk.Button(new_window, text="Colors", command=choose_color,width=7
colors.pack()

def orange_xy():
    b.set_group(selected_value_gid,'xy',[0.550,0.400])
orange_b = tk.Button(new_window, text="Pre-Set Orange", command=orange_xy,b
orange_b.pack()
```

```
def party_window():
    new_window = Tk()
    new_window.title("Party Window")
    party_label = ttk.Label(new_window, text="Select a Room:")
    party_label.pack()
    rooms = {'Salman Bedroom': ['1', '2', '4', '7'],
                'Living Room': ['5', '6'],
                'Ahmed Bedroom': ['9']
                }
    combo = ttk.Combobox(new_window, values=list(rooms.keys()))
    combo.pack()

    def get_selected_value_party_mode():

        global selected_value
        selected_value = combo.get()
        print("Selected Value:", selected_value)
        global get_group_sv
        get_group_sv = b.get_group(selected_value) #get_group seleceted value
        print(get_group_sv)
        global selected_value_lid #light_id
        selected_value_lid= [int(light_id) for light_id in get_group_sv['lights
        print(selected_value_lid)
    button = tk.Button(new_window, text="Confrim", command=get_selected_value_p
    button.pack()



    # Function to start the party
    def party():
        start = time.time()
        global x
        x = 1
        #duration = 5
        duration = simpledialog.askstring("Input", "Enter duration of حفلة مود
        duration=int(duration)
        while x > 0:
            for light in get_group_sv:
                b.set_light(selected_value_lid,'on',True)
                b.set_light(selected_value_lid, 'bri', 254)
                b.set_light(selected_value_lid, 'xy', [random.random(), random.
            elapsed_time = time.time() - start
            print(elapsed_time)
            if elapsed_time >= duration:
                x=0
            # Create a button to start the party
    start_party = tk.Button(new_window, text="ابدأ الحفلة!", command=party)
    start_party.pack()



#OVERVIEW FUNCTION (QUICK STATUS)
def status(g_id):
    if b.get_group(g_id,'on')==True:
```

```
        bri=b.get_group(g_id,'bri')
        status = f'Lights are on, Brightness is {bri}'
        return status

    else:
        status='Lights are off'
        return status

#MAIN WINDOW
root=Tk()
root.title("Smart Home Control")

                    #OVERVIEW SECTION
#OVERVIEW LABEL
overview_label = tk.Label(root, text="Overview",fg='#5f43b2')
overview_label.grid(row=0, column=1)

#IMG 1
image_path = '/Users/salman/Desktop/Github/Smart Home/IMG_6384.jpg'

original_image = Image.open(image_path)
corrected_image = original_image.rotate(270, expand=True)
new_width = 50
new_height = 50
resized_image = corrected_image.resize((new_width, new_height), Image.ANTIALIAS

photo = ImageTk.PhotoImage(resized_image)

label_image = tk.Label(root, image=photo)
label_image.grid(row=2, column=0,padx=10,pady=10)

#OVERVIEW STATUS SALMAN
sb_status_label = tk.Label(root, text="Salman Bedroom",fg='#5f43b2')
sb_status_label.grid(row=3, column=0)

status_label = tk.Label(root, text="", width=30)
status_label.grid(row=4,column=0)

def update_status():
    get_status = status(1)
    status_label.config(text=get_status)
    root.after(1000, update_status)
update_status()



#IMG 2
image_path2 = '/Users/salman/Desktop/Github/Smart Home/IMG_9494.jpg'

original_image2 = Image.open(image_path2)
corrected_image2 = original_image2.rotate(270, expand=True)
new_width = 50
new_height = 50
resized_image2 = corrected_image2.resize((new_width, new_height), Image.ANTIALI

photo2 = ImageTk.PhotoImage(resized_image2)

label_image2 = tk.Label(root, image=photo2)
label_image2.grid(row=2, column=1,padx=10,pady=10)
#OVERVIEW STATUS AHMED
```

```
ab_status_label = tk.Label(root, text="Ahmed Bedroom",fg='#5f43b2')
ab_status_label.grid(row=3, column=1)

status_label_ab = tk.Label(root, text="", width=30)
status_label_ab.grid(row=4,column=1)


def update_status_ab():
    get_status_ab = status(4)
    status_label_ab.config(text=get_status_ab)
    root.after(1000, update_status_ab)
update_status_ab()


#IMG 3
image_path3 = '/Users/salman/Desktop/Github/Smart Home/lito.JPG'

original_image3 = Image.open(image_path3)
corrected_image3 = original_image3.rotate(360, expand=True)

new_width = 50
new_height = 50
resized_image3 = corrected_image3.resize((new_width, new_height), Image.ANTIALI

photo3 = ImageTk.PhotoImage(resized_image3)

label_image3 = tk.Label(root, image=photo3)
label_image3.grid(row=2, column=3,padx=10,pady=10)

#OVERVIEW STATUS LIVING ROOM
lv_status_label = tk.Label(root, text="Living Room (Ales Room)",fg='#5f43b2')
lv_status_label.grid(row=3, column=3)

status_label_lv = tk.Label(root, text="", width=30)
status_label_lv.grid(row=4,column=3)

def update_status_lv():
    get_status_lv = status(3)
    status_label_lv.config(text=get_status_lv)
    root.after(1000, update_status_lv)
update_status_lv()


#LIVING ROOM LABELS
lv_label = tk.Label(root, text="Living Room", fg='#5f43b2')
lv_label.grid(row=5, column=0)

lv_on_b = tk.Button(root, text="ON", command=livingroom_on,width=7)
lv_on_b.grid(row=6, column=1)

lv_of_b = tk.Button(root, text="OFF", command=livingroom_off,width=7)
lv_of_b.grid(row=7, column=1)

lv_dim_b = tk.Button(root, text="Dimmer", command=livingroom_dimmer,width=7)
lv_dim_b.grid(row=8,column=1)

#SALMAN BEDROOM LABELS
sb_label = tk.Label(root, text="Salmans Bedroom",fg='#5f43b2')
sb_label.grid(row=9, column=0)
```

```
sb_on_b = tk.Button(root, text="ON", command=salmanroom_on,width=7)
sb_on_b.grid(row=10, column=1)

sb_of_b = tk.Button(root, text="OFF", command=salmanroom_off,width=7)
sb_of_b.grid(row=11, column=1)

sb_dim_b = tk.Button(root, text="Dimmer", command=salman_dimmer,width=7)
sb_dim_b.grid(row=12, column=1)


#AHMED BEDROOM LABELS
ab_label = tk.Label(root, text="Ahmed Bedroom", fg='#5f43b2')
ab_label.grid(row=14, column=0)

ab_on_b = tk.Button(root, text="ON", command=ahmedroom_on,width=7)
ab_on_b.grid(row=15, column=1)

ab_of_b = tk.Button(root, text="OFF", command=ahmedroom_off,width=7)
ab_of_b.grid(row=16, column=1)

ab_dim_b = tk.Button(root, text="Dimmer", command=ahmed_dimmer,width=7)
ab_dim_b.grid(row=17,column=1)

#ALL LIGHTS LABELS
all_label = tk.Label(root, text="All lights",fg='#5f43b2')
all_label.grid(row=18, column=0)

all_on_b = tk.Button(root, text="ON", command=all_on,width=7)
all_on_b.grid(row=19, column=1)

all_of_b = tk.Button(root, text="OFF", command=all_off,width=7)
all_of_b.grid(row=20, column=1)

all_dim_b = tk.Button(root, text="Dimmer", command=all_dimmer,width=7)
all_dim_b.grid(row=21, column=1)

#COLORS LABELS
colors_label = tk.Label(root, text="Change Colors",fg='#5f43b2')
colors_label.grid(row=22, column=0)


colors_b = tk.Button(root, text="Colors", command=color_window,width=7)
colors_b.grid(row=23, column=1)

party_b = tk.Button(root, text="حفلة مود", command=party_window,width=7)
party_b.grid(row=24, column=1)

root.mainloop()
```