## Front End CODE EXPLANAITON

**Frame:**

This frame is used to create the frames on the window

**Label:**

This keyword is used to display sometext on the screen.

**Pack():**

Fix the place , height, width of the frame on the window

**grid():**

This fixes the **place** of the **label or button** on the gui window. How much it will be padding from x axis and how much from y and many more things.

1. Created five objects as said by the project and sequence was followed by the series of devices according to the roll no. last digit.

This process was easy and it don’t need any further explanation. Simple keyword are used of the python language.

## Created a main frame for the whole GUI:

The idea behind the main Frame was to encapsulate all next coming frames inside this main frame. All other frames will come inside of this frame.

Tkinter library is commonly used for the gui and it’s imported as asked by the project.

import tkinter as tk

Here the tkinter is imported with the alias tk so it can be called be reffered more easily in the code with this object tk. Then we created the object for the tkinter function Tk() with the tkinter object tk.

Mainwin=tk.Tk()

Here we called the tkinter function tk() for gui with the tkinter object tk and stored it in the mainwin object so the function can be easly accessed with this object.

### Code for the main frame:

#creating the main frame

    wf=tk.Frame(mianWin,bd=2,border=5,relief="groove",bg="gray")

    wf.pack(side="top",fill="both")

here wf(whole frame)

The function is called by the tkinter alias tk. The function Frame is part of the tkinter library so called it with it’s object/alias.

**mainwin:** Mainwin will be main object which will be referencing the window of the gui. It will be the container for the all the widgets for the this frame. All the information on this frame will come inside of the mainwin window.

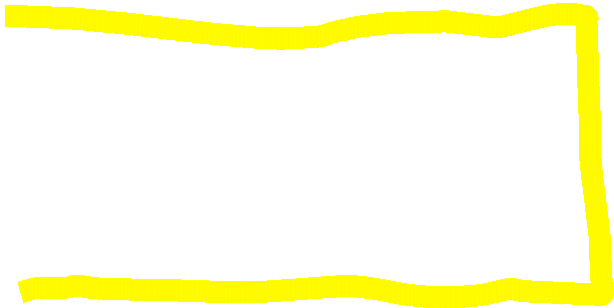
**Border** = this value is representing the outer border of the frame.

**Relief =”groove”:** This parameter is used to set the border style for the frame. Here groove.

**Pack()**

Then we called the pack function on the frame with it’s object wf. The pack function is setting the place for the frame on the gui window. Here we set the place at top and asked it to fill all the horizontal space that comes in front of it.

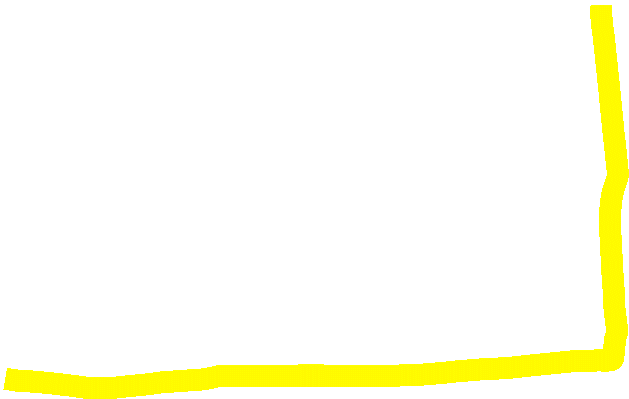
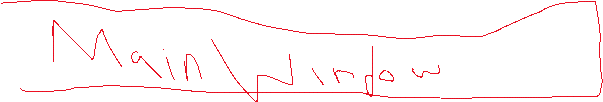
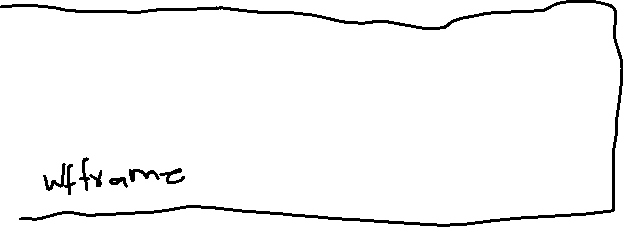
This Yellow line is the gui window and the wf is the main frame on the window.



In all other frame we will pass the wf object to all frames creation so that they come inside of this frame. The size of the wf will grow with the text that will come inside

## Then inside of this main frame we created another frame : Heading and all the objects and created their corresponding toggle buttons. All the frames will come inside of the main wf frame.

When we created the heading frame, this happens.



## When we will be creating further frames with stickey=”ew” flag then that frame will comes on the new line. Right down the already frame on the window.

## I have created two frames for the all devices which have toggle button and their toggle button. One frame is covering the object label(displaying value on the gui) and second value their toggle button.



butt5=tk.Button(toggle5,relief="solid",text="Toggle this",command=lambda:label5.config(text="Plug :{1} ,  {2} consumption  ".format(plug3.toggleSwitch() , plug3.get\_switch(),plug3.get\_rate())))

now what this line of code is doing

firstly this is displaying button on the window and the text of the button is within the text keyword.

Command is done when the button is pressed, that function is checking the button check automatically. And Lambda is checking if the button is pressed then do that command with will be following it.

The further statement will access the label 5 text (remember that the label is use to display text on the screen and it has an object to access that label) and use the config function to change the text of that label and give it new value. The new values are access are toggled and accessed by the respective object )

The keyword binding is binding the button with some specific action on that button.

For example:

**label.bind(‘if the something happened on that label e.g <buttonpressed>’, call this function e.g count).** if the label has perfomed some action then do that that will call the respective function.