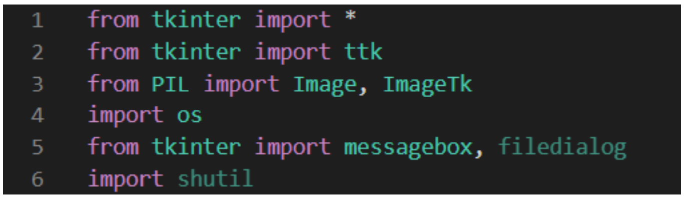
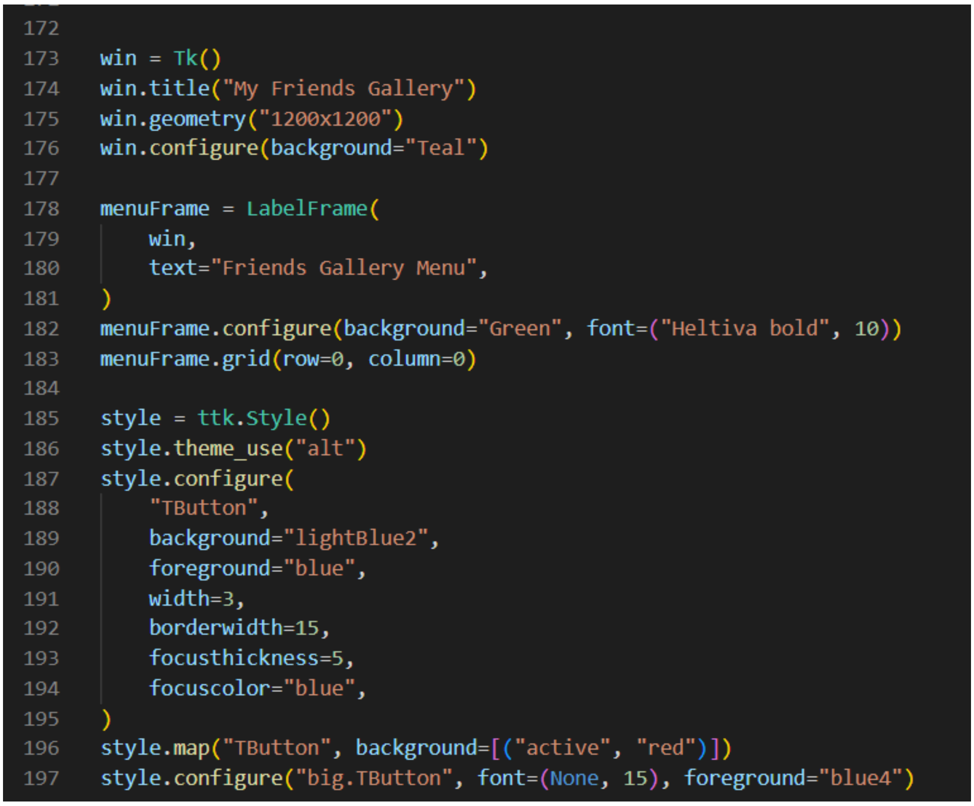
Ayra Ahmed w1947450



```python
from tkinter import *
from tkinter import ttk
from PIL import Image, ImageTk
import os
from tkinter import messagebox, filedialog
import shutil
```

This will import tkinter, ttkk, to get the window.

It will also import Image using PIL, which will show the pictures.

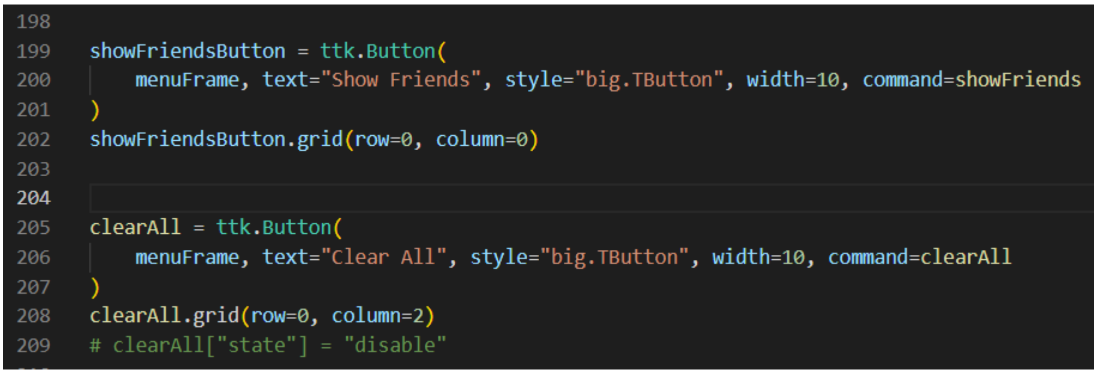


```python

win = Tk()
win.title("My Friends Gallery")
win.geometry("1200x1200")
win.configure(background="Teal")

menuFrame = LabelFrame(
    win,
    text="Friends Gallery Menu",
)
menuFrame.configure(background="Green", font=("Heltiva bold", 10))
menuFrame.grid(row=0, column=0)

style = ttk.Style()
style.theme_use("alt")
style.configure(
    "TButton",
    background="lightBlue2",
    foreground="blue",
    width=3,
    borderwidth=15,
    focusthickness=5,
    focuscolor="blue",
)
style.map("TButton", background=[("active", "red")])
style.configure("big.TButton", font=(None, 15), foreground="blue4")
```

This will make the window called My Friends Gallery, this window will have a size of 1200 x 1200 and will have a teal background

On that window there will be a frame, which has a green background called Friends Gallery Menu. This will be where the buttons appear, for the user to press

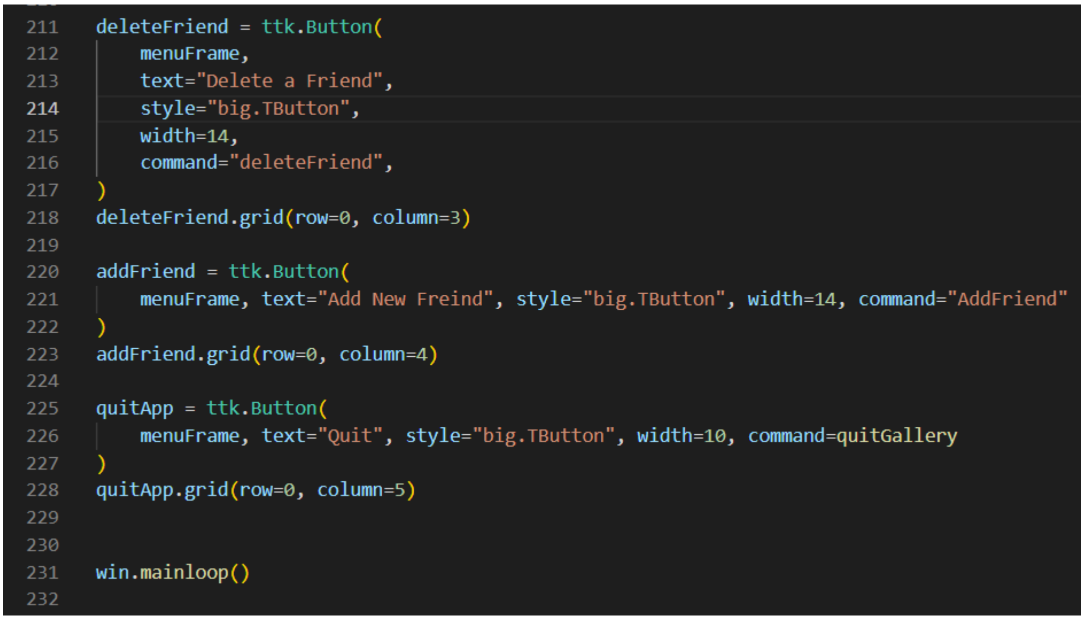
This is the code we use to make the buttons, by importing ttk.style , each button will have a lightblue background and the writing will be in blue



```python

showFriendsButton = ttk.Button(
    menuFrame, text="Show Friends", style="big.TButton", width=10, command=showFriends
)
showFriendsButton.grid(row=0, column=0)


clearAll = ttk.Button(
    menuFrame, text="Clear All", style="big.TButton", width=10, command=clearAll
)
clearAll.grid(row=0, column=2)
# clearAll["state"] = "disable"
```

The first button is show friends. This will show the friends when pressed.

The second button is clear All, this will clear the friends when pressed.



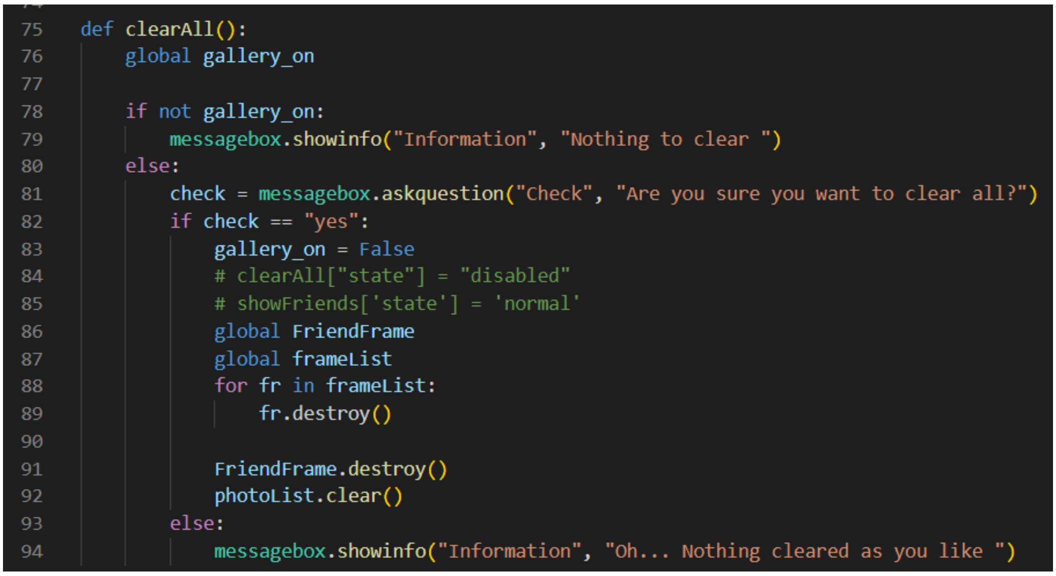
```python
deleteFriend = ttk.Button(
    menuFrame,
    text="Delete a Friend",
    style="big.TButton",
    width=14,
    command="deleteFriend",
)
deleteFriend.grid(row=0, column=3)

addFriend = ttk.Button(
    menuFrame, text="Add New Freind", style="big.TButton", width=14, command="AddFriend"
)
addFriend.grid(row=0, column=4)

quitApp = ttk.Button(
    menuFrame, text="Quit", style="big.TButton", width=10, command=quitGallery
)
quitApp.grid(row=0, column=5)


win.mainloop()
```

the third button is delete friend this will allow you delete a friend from the list

the fourth button is add friend, this will allow you to add a friend from the list.

The last button is quit, this will clear the window, in other words quit the my friends gallery window.

The win.mainloop(), allows python to run the Tkinter event loop.

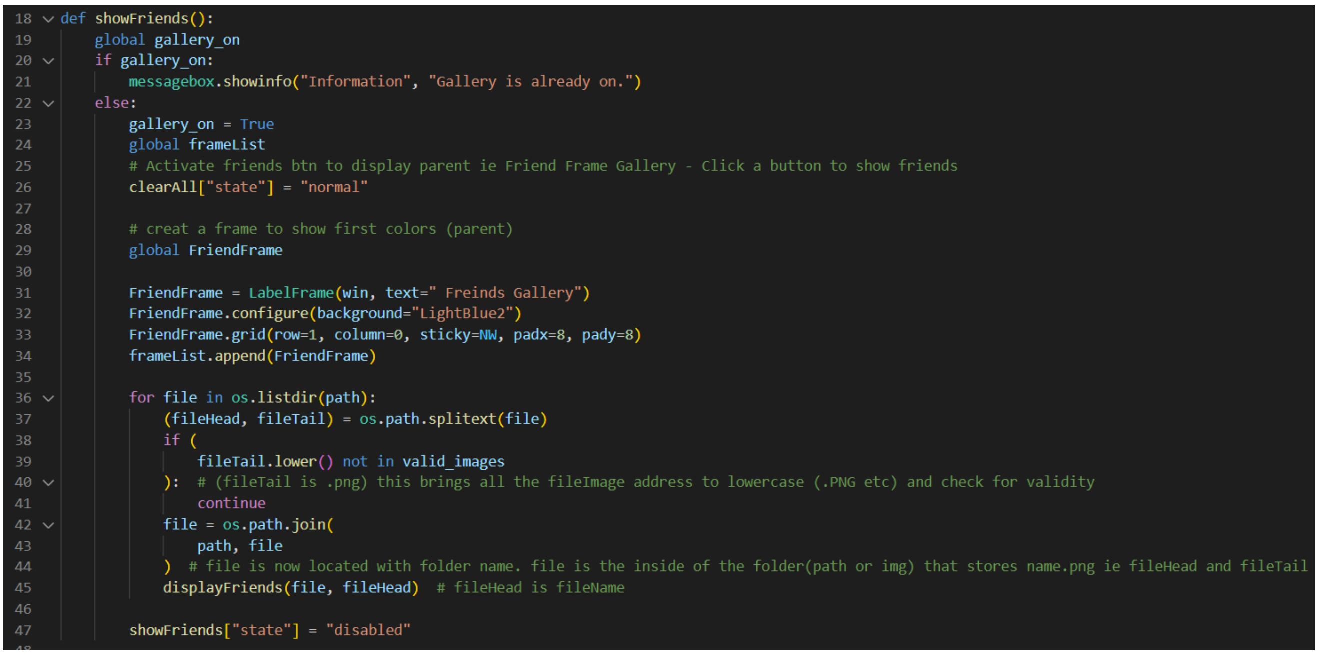


```python
def clearAll():
    global gallery_on

    if not gallery_on:
        messagebox.showinfo("Information", "Nothing to clear ")
    else:
        check = messagebox.askquestion("Check", "Are you sure you want to clear all?")
        if check == "yes":
            gallery_on = False
            # clearAll["state"] = "disabled"
            # showFriends['state'] = 'normal'
            global FriendFrame
            global frameList
            for fr in frameList:
                fr.destroy()

            FriendFrame.destroy()
            photoList.clear()
        else:
            messagebox.showinfo("Information", "Oh... Nothing cleared as you like ")
```

This function shows a message box when if the list of friends is not being shown.

However, if the list of friends is on, then it will show a question box to ask if the user wants to clear the gallery. The user will have to press yes to clear and if the user pressed no then it will show an information box saying that nothing is cleared.

Ayra Ahmed w1947450



```
def showFriends():
    global gallery_on
    if gallery_on:
        messagebox.showinfo("Information", "Gallery is already on.")
    else:
        gallery_on = True
        global frameList
        # Activate friends btn to display parent ie Friend Frame Gallery - Click a button to show friends
        clearAll["state"] = "normal"

        # creat a frame to show first colors (parent)
        global FriendFrame

        FriendFrame = LabelFrame(win, text=" Freinds Gallery")
        FriendFrame.configure(background="LightBlue2")
        FriendFrame.grid(row=1, column=0, sticky=NW, padx=8, pady=8)
        frameList.append(FriendFrame)

        for file in os.listdir(path):
            (fileHead, fileTail) = os.path.splitext(file)
            if (
                fileTail.lower() not in valid_images
            ):  # (fileTail is .png) this brings all the fileImage address to lowercase (.PNG etc) and check for validity
                continue
            file = os.path.join(
                path, file
            )  # file is now located with folder name. file is the inside of the folder(path or img) that stores name.png ie fileHead and fileTail
            displayFriends(file, fileHead)  # fileHead is fileName

        showFriends["state"] = "disabled"
```

This function will show that if the list of friends is on and the show friends button is pressed a message box will appear saying that the gallery is already on. And it will disable the clear all button.

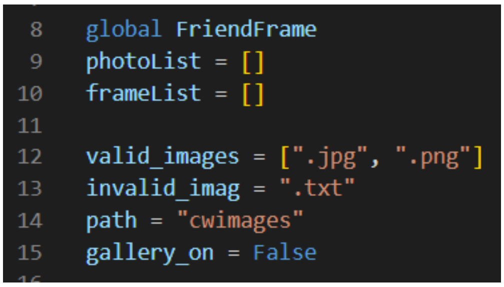
However, when you pressed the show friends button again to activate the list of friends, the clear all button will be activated again.

The friend frame, is created, this is where the list of friends will appear as buttons. This frame is also known as the parent frame.

The for loop allows for the images to appear of the friends. fileTail is .png, this brings all the fileImage address to lowercase and checks for validity

File = os.path.join (path,file), this show that the file is now located with folder name, the file is the inside of the folder (path or img) that stores name.png ie fileHead and fileTail. fileHead is fileName.

Then the show friends button will be disabled when the list of friends appear.

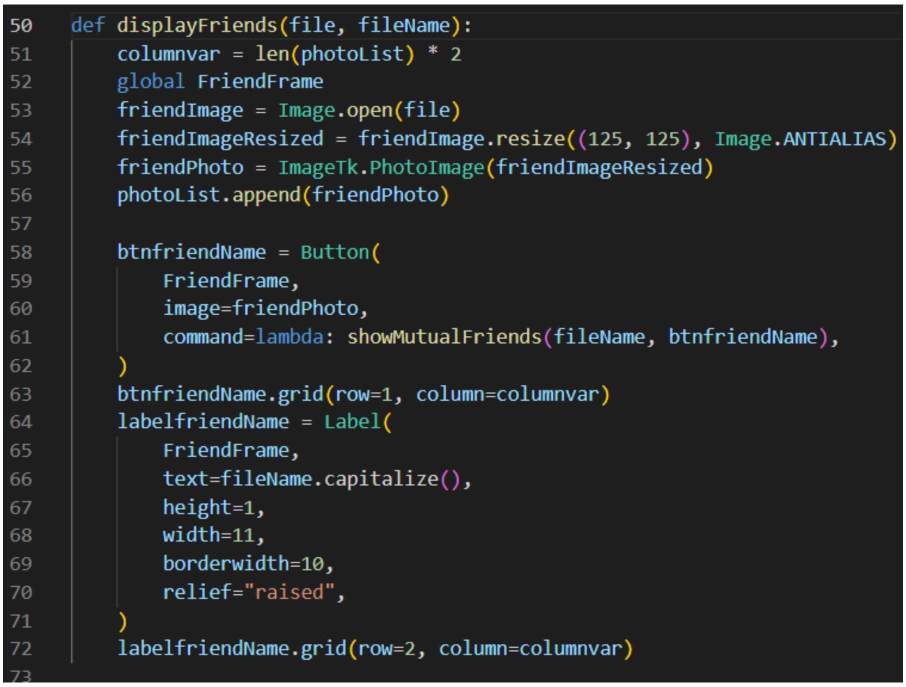


```
global FriendFrame
photoList = []
frameList = []

valid_images = [".jpg", ".png"]
invalid_imag = ".txt"
path = "cwimages"
gallery_on = False
```

The global keyword is used to create or declare a global variable inside a function.

The valid images and invalid images shows what pictures we are used and what pictures we are not using.

Path shows where the file is and the name of the file.



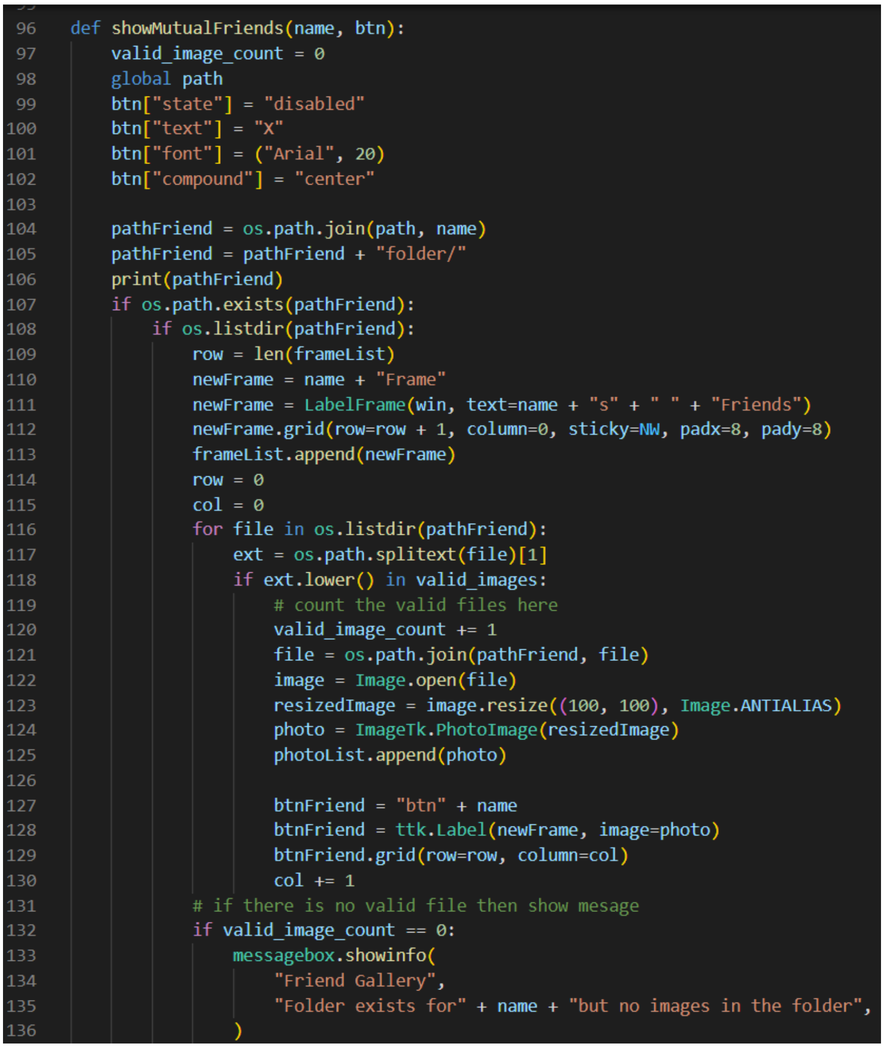
```
def displayFriends(file, fileName):
    columnvar = len(photoList) * 2
    global FriendFrame
    friendImage = Image.open(file)
    friendImageResized = friendImage.resize((125, 125), Image.ANTIALIAS)
    friendPhoto = ImageTk.PhotoImage(friendImageResized)
    photoList.append(friendPhoto)

    btnfriendName = Button(
        FriendFrame,
        image=friendPhoto,
        command=lambda: showMutualFriends(fileName, btnfriendName),
    )
    btnfriendName.grid(row=1, column=columnvar)
    labelfriendName = Label(
        FriendFrame,
        text=fileName.capitalize(),
        height=1,
        width=11,
        borderwidth=10,
        relief="raised",
    )
    labelfriendName.grid(row=2, column=columnvar)
```

This function will make the images appear in a row and tells python where the images are.

Then it will create buttons for the images

Lambda = evaluates and returns only one expression.

Then it will also show the names of each picture.

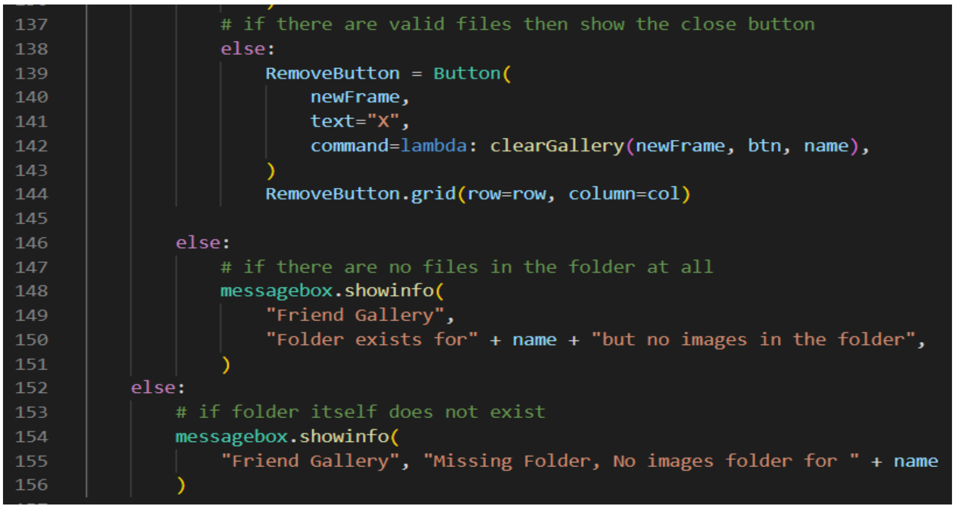


```
def showMutualFriends(name, btn):
    valid_image_count = 0
    global path
    btn["state"] = "disabled"
    btn["text"] = "X"
    btn["font"] = ("Arial", 20)
    btn["compound"] = "center"

    pathFriend = os.path.join(path, name)
    pathFriend = pathFriend + "folder/"
    print(pathFriend)
    if os.path.exists(pathFriend):
        if os.listdir(pathFriend):
            row = len(frameList)
            newFrame = name + "Frame"
            newFrame = LabelFrame(win, text=name + "s" + " " + "Friends")
            newFrame.grid(row=row + 1, column=0, sticky=NW, padx=8, pady=8)
            frameList.append(newFrame)
            row = 0
            col = 0
            for file in os.listdir(pathFriend):
                ext = os.path.splitext(file)[1]
                if ext.lower() in valid_images:
                    # count the valid files here
                    valid_image_count += 1
                    file = os.path.join(pathFriend, file)
                    image = Image.open(file)
                    resizedImage = image.resize((100, 100), Image.ANTIALIAS)
                    photo = ImageTk.PhotoImage(resizedImage)
                    photoList.append(photo)

                    btnFriend = "btn" + name
                    btnFriend = ttk.Label(newFrame, image=photo)
                    btnFriend.grid(row=row, column=col)
                    col += 1
            # if there is no valid file then show mesage
            if valid_image_count == 0:
                messagebox.showinfo(
                    "Friend Gallery",
                    "Folder exists for" + name + "but no images in the folder",
                )
```

This function will show each friend in the friends list, for example it will show Adam's friends and Alex's friends.

The path friend shows where the images are in the file and how to arrange it. in this case the images will appear in a row.

Then it will count the valid image files and if there is no valid image files it will show a message information box saying that this specific person has folder but no images.

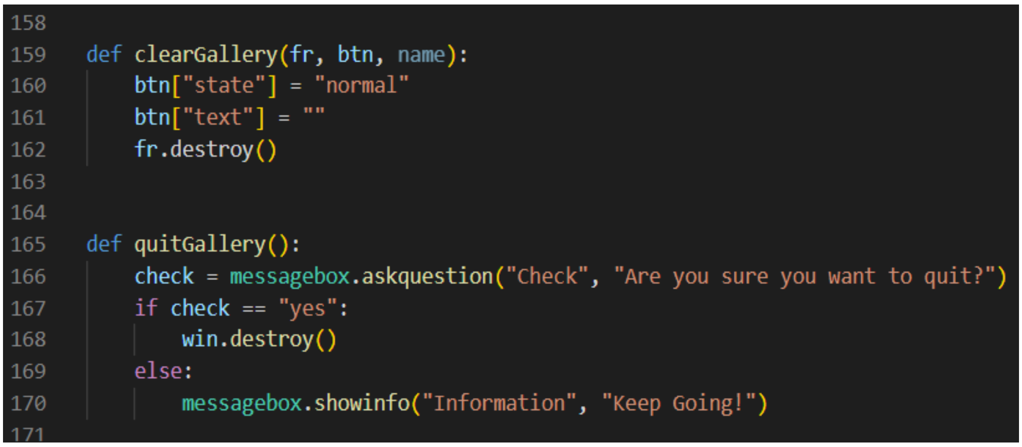


```
            # if there are valid files then show the close button
            else:
                RemoveButton = Button(
                    newFrame,
                    text="X",
                    command=lambda: clearGallery(newFrame, btn, name),
                )
                RemoveButton.grid(row=row, column=col)

        else:
            # if there are no files in the folder at all
            messagebox.showinfo(
                "Friend Gallery",
                "Folder exists for" + name + "but no images in the folder",
            )
    else:
        # if folder itself does not exist
        messagebox.showinfo(
            "Friend Gallery", "Missing Folder, No images folder for " + name
        )
```

else if there are valid files then show the close button

else if there are no files in the folder at all a message information box will appear

else if the folder itself does not exist then a message information box will appear saying that this person is missing a folder.



```
def clearGallery(fr, btn, name):
    btn["state"] = "normal"
    btn["text"] = ""
    fr.destroy()


def quitGallery():
    check = messagebox.askquestion("Check", "Are you sure you want to quit?")
    if check == "yes":
        win.destroy()
    else:
        messagebox.showinfo("Information", "Keep Going!")
```

This function clears the gallery and returns the button back to normal

The quit function asks the user if they want to quit and if pressed yes then it will quit the window and if pressed no then an information box will appear allowing the user to continue using window.