# In every route which requires login , just put if logged\_in:

# Logout button that send request to /logout

from flask import Flask, render\_template, url\_for

from flask import jsonify, request

from flask import flash, redirect,abort

import datetime as d

app = Flask(\_\_name\_\_)

lastOffTime = d.datetime.now()

netlitres = 0

prevtank1 = 10

diff = 0

x = 0

logged\_in = False

power =0

tank1data = 40

tank2data = 40

current =  "OFF"

option = ""

ontime = 0

stat1 = ""

stat2 = ""

stat3 = ""

'''@app.route('/',  methods=['GET'])

def index():

    global tank1\_level

    return render\_template('index\_gauge.html')  '''

@app.route('/')

def home():

    if logged\_in:

        return render\_template('home.html')

    else:

        return redirect('/login')

    lastOffTime = d.datetime.now()

@app.route('/login', methods=['GET'])

def login\_page():

    return render\_template('login.html')

@app.route('/logout', methods=['GET'])

def logout():

    global logged\_in

    logged\_in = False

    return render\_template('login.html')

@app.route('/login', methods=['POST'])

def check\_login():

    global logged\_in

    if request.form['password'] == 'password' and request.form['username'] == 'admin':

        logged\_in = True

        return redirect('/')

    else:

        return render\_template('login.html')

@app.route('/water', methods=['GET'])

def water():

    if logged\_in:

        return render\_template('index\_gauge.html')

    else:

        return redirect('/login')

@app.route('/energy', methods=['GET'])

def energy():

    if logged\_in:

        return render\_template('energy.html')

    else:

        return redirect('/login')

'''@app.route('/power/<int:p>', methods=['GET'])

def power():

    global power

    power = p

    return 'ok' '''

@app.route('/deptho/<int:depth\_cm1>', methods=['GET'])

def show\_post1(depth\_cm1):

    global tank1data

    global netlitres

    global prevtank1

    global diff

    if depth\_cm1 < prevtank1:

        diff = (prevtank1 - depth\_cm1)

    tank1data = depth\_cm1

    prevtank1 = depth\_cm1

    netlitres = netlitres + diff

    return 'ok'

@app.route('/power/<int:p>', methods=['GET'])

def power(p):

    global power

    power = p

    return 'ok'

@app.route('/change/<string:switch>', methods=['GET','POST'])

def change(switch):

    global option

    option = switch

    return option

@app.route('/stat/<int:p>', methods=['GET'])

def status(p):

    global ontime

    global tank1data

    global tank2data

    global current

    global option

    global stat1,stat2,stat3

    global power

    power = p

    s1 =""

    s2 =""

    s3 = ""

    if tank1data<20 and tank2data>20 and option == "auto":

        s1 = "xyz"

        stat1="a"

        current = "On"

    elif option == "on":

        s2 = "abc"

        stat2 = "a"

    elif option == "off":

        s2 = ""

        stat2 = ""

    elif option == "a2on":

        s3 = "pqr"

        stat3 = "a"

    elif option == "a2off":

        s3 = ""

        stat3 = ""

    elif option == "a1on":

        s1 = "xyz"

        stat1 = "a"

        current = "On"

    elif option == "a1off":

        s1 = ""

        stat1  = ""

        current = "Off"

    pfinal = s1 + s2 + s3

    if stat1 =="a":

        pfinal = pfinal + "xyz"

    if stat2 == "a":

        pfinal = pfinal + "abc"

    if stat3 == "a":

        pfinal = pfinal + "pqr"

    final = pfinal + "$"

    return final

@app.route('/depths/<int:depth\_cm2>', methods=['GET'])

def show\_post2(depth\_cm2):

    global tank2data

    tank2data = depth\_cm2

    return 'ok'

@app.route('/return\_global', methods=['GET'])

def return\_global():

    global tank1data

    global tank2data

    global current

    global ontime

    global lastOffTime

    global netlitres

    global x

    global power

    x = x+ 1

    if current == "OFF":

        lastOffTime = d.datetime.now()

    elif current == "On":

        if  0==0:

            f= 0

            diff = d.datetime.now() - lastOffTime

            lastOffTime = d.datetime.now()

            f = diff.microseconds

            ontime = ontime + f

    return jsonify(tank1 = tank1data , tank2 = tank2data, stat = current, time = ontime/1000000, net= netlitres)

if \_\_name\_\_ == "\_\_main\_\_":

    app.run(host='0.0.0.0', port=8080, debug=True)