```python
#main.py | Robin Forestier | 2021 / 2022

from flask import Flask, g, render_template, request, session, url_for, redirect
import time
import datetime
import threading
import csv
import os
import logging

import RPi.GPIO as GPIO

logging.basicConfig(filename="log.log", format='%(asctime)s - %(levelname)s - %(message)s', level=logging.DEBUG)

GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)

#GPIO 0 to 11 in output mode at 0
for i in range(12):
    GPIO.setup(i, GPIO.OUT, initial=GPIO.LOW)

app = Flask(__name__)
app.secret_key = 'somesecretkeythatonlyishouldknow'
app.session_cookie_name = 'MyBeautifulCookies'

authorize_ip = ["localhost", "127.0.0.1", "172.16.32.133"]

buttonSts_p1 = ["/static/img/img_off.png"] * 8
buttonSts_p2 = ["/static/img/img_off.png"] * 8
color = ["#333333"] * 8
warning = ""

class User:
    def __init__(self, id, username, password):
        self.id = id
        self.username = username
        self.password = password

    def __repr__(self):
        return f'<User: {self.username}>'

users = []
users.append(User(id=1, username='elo', password='elo'))
users.append(User(id=2, username='admin', password='admin'))
users.append(User(id=3, username='local', password='local'))

def gpio_modif():

    for i in range(8):
        if buttonSts_p1[i] == "/static/img/img_off.png":
            #OFF
            GPIO.output(i, 0)
        else:
            #ON
            GPIO.output(i, 1)
            #time.sleep(1)      #security time for fuses

def getTime():
    t = time.localtime()
    current_time = time.strftime("%H:%M", t)

    return current_time

@app.before_request
def before_request():
    g.user = None

    ip = request.environ.get('HTTP_X_REAL_IP', request.remote_addr)

    if 'user_id' in session :
        #logging.debug("reconnect " + str(ip))
```

```
72          user = [x for x in users if x.id == session['user_id']][0]
73          g.user = user
74
75      else :
76          for i in authorize_ip:
77              if ip == i:
78                  if 'user_id' in session:
79                      #logging.debug("reconect as local " + str(ip))
80                      user = [x for x in users if x.id == session['user_id']][0]
81                      g.user = user
82
83                  else :
84                      logging.info("creat local user" + str(ip))
85                      session.pop('user_id', None)
86                      session['user_id'] = 3
87                      return redirect(url_for('page1'))
88
89  @app.route("/", methods=['POST', 'GET'])
90  def login():
91      current_time = getTime()
92
93      if request.method == 'POST':
94          session.pop('user_id', None)
95
96          username = request.form['username']
97          password = request.form['password']
98
99          ip = request.environ.get('HTTP_X_REAL_IP', request.remote_addr)
100
101         try:
102             user = [x for x in users if x.username == username][0]
103             try:
104                 if user and user.password == password:
105                     session['user_id'] = user.id
106                     logging.info("New login username : " + username + " ip : ",
                        str(ip))
107                     return redirect(url_for('page1'))
108             except:
109                 logging.warning("bad password : " + username + " ip : " + str(ip))
110                 return redirect(url_for('login'))
111         except:
112             logging.warning("bad username : " + username + " ip : " + str(ip))
113             return redirect(url_for('login'))
114
115     return render_template("login.html", time=current_time, warning=warning)
116
117 @app.route("/page1", methods = ['POST', 'GET'])
118 def page1():
119     current_time = getTime()
120
121     if not g.user:
122         return redirect(url_for('login'))
123
124     if all(elem == "/static/img/img_on.png" for elem in buttonSts_p1):
125         buttonSts_p2[0] = "/static/img/img_on.png"
126     else:
127         buttonSts_p2[0] = "/static/img/img_off.png"
128
129     if request.method == 'POST':
130         if request.form['button_p1'] == '1':
131             if buttonSts_p1[0] == "/static/img/img_on.png":
132                 buttonSts_p1[0] = "/static/img/img_off.png"
133                 color[0] = "#333333"
134             else:
135                 buttonSts_p1[0] = "/static/img/img_on.png"
136                 color[0] = "#FFFFFF"
137         elif request.form['button_p1'] == '2':
138             if buttonSts_p1[1] == "/static/img/img_on.png":
139                 buttonSts_p1[1] = "/static/img/img_off.png"
140                 color[1] = "#333333"
141             else:
142                 buttonSts_p1[1] = "/static/img/img_on.png"
```

```
143                 color[1] = "#FFFFFF"
144         elif request.form['button_p1'] == '3':
145             if buttonSts_p1[2] == "/static/img/img_on.png":
146                 buttonSts_p1[2] = "/static/img/img_off.png"
147                 color[2] = "#333333"
148             else:
149                 buttonSts_p1[2] = "/static/img/img_on.png"
150                 color[2] = "#FFFFFF"
151         elif request.form['button_p1'] == '4':
152             if buttonSts_p1[3] == "/static/img/img_on.png":
153                 buttonSts_p1[3] = "/static/img/img_off.png"
154                 color[3] = "#333333"
155             else:
156                 buttonSts_p1[3] = "/static/img/img_on.png"
157                 color[3] = "#FFFFFF"
158         elif request.form['button_p1'] == '5':
159             if buttonSts_p1[4] == "/static/img/img_on.png":
160                 buttonSts_p1[4] = "/static/img/img_off.png"
161                 color[4] = "#333333"
162             else:
163                 buttonSts_p1[4] = "/static/img/img_on.png"
164                 color[4] = "#FFFFFF"
165         elif request.form['button_p1'] == '6':
166             if buttonSts_p1[5] == "/static/img/img_on.png":
167                 buttonSts_p1[5] = "/static/img/img_off.png"
168                 color[5] = "#333333"
169             else:
170                 buttonSts_p1[5] = "/static/img/img_on.png"
171                 color[5] = "#FFFFFF"
172         elif request.form['button_p1'] == '7':
173             if buttonSts_p1[6] == "/static/img/img_on.png":
174                 buttonSts_p1[6] = "/static/img/img_off.png"
175                 color[6] = "#333333"
176             else:
177                 buttonSts_p1[6] = "/static/img/img_on.png"
178                 color[6] = "#FFFFFF"
179         elif request.form['button_p1'] == '8':
180             if buttonSts_p1[7] == "/static/img/img_on.png":
181                 buttonSts_p1[7] = "/static/img/img_off.png"
182                 color[7] = "#333333"
183             else:
184                 buttonSts_p1[7] = "/static/img/img_on.png"
185                 color[7] = "#FFFFFF"
186         elif request.form['button_p1'] == 'page_2':
187             return redirect(url_for('page2'))
188         else:
189             pass
190 
191         gpio_modif()
192 
193     return render_template('page1.html', button=buttonSts_p1, color=color,
    time=current_time, warning=warning)
194 
195 @app.route("/page2", methods = ['POST', 'GET'])
196 def page2():
197     current_time = getTime()
198 
199     if not g.user:
200         return redirect(url_for('login'))
201 
202     if request.method == 'POST':
203         if request.form['button_p1'] == '1':
204             buttonSts_p2[0] = "/static/img/img_on.png"
205             for i in range(8):
206                 buttonSts_p1[i] = "/static/img/img_on.png"
207                 color[i] = "#FFFFFF"
208         elif request.form['button_p1'] == '2':
209             buttonSts_p2[1] = "/static/img/img_off.png"
210             for i in range(8):
211                 buttonSts_p1[i] = "/static/img/img_off.png"
212                 buttonSts_p2[i] = "/static/img/img_off.png"
213                 color[i] = "#333333"
```

```
214             elif request.form['button_p1'] == '3':
215                 buttonSts_p2[0] = "/static/img/img_off.png"
216                 buttonSts_p2[1] = "/static/img/img_off.png"
217 
218                 for i in range(0, 8, 2):
219                     color[i] = "#FFFFFF"
220                     color[i + 1] = "#333333"
221 
222                     buttonSts_p1[i] = "/static/img/img_on.png"
223                     buttonSts_p1[i + 1] = "/static/img/img_off.png"
224             elif request.form['button_p1'] == '4':
225                 buttonSts_p2[0] = "/static/img/img_off.png"
226                 buttonSts_p2[1] = "/static/img/img_off.png"
227 
228                 for i in range(0, 8):
229                     if i < 3:
230                         color[i] = "#FFFFFF"
231                         buttonSts_p1[i] = "/static/img/img_on.png"
232                     else :
233                         color[i] = "#333333"
234                         buttonSts_p1[i] = "/static/img/img_off.png"
235             elif request.form['button_p1'] == '5':
236                 buttonSts_p2[0] = "/static/img/img_off.png"
237                 buttonSts_p2[1] = "/static/img/img_off.png"
238 
239                 for i in range(0, 8):
240                     if i > 3 and i != 7:
241                         color[i] = "#FFFFFF"
242                         buttonSts_p1[i] = "/static/img/img_on.png"
243                     else:
244                         color[i] = "#333333"
245                         buttonSts_p1[i] = "/static/img/img_off.png"
246 
247             elif request.form['button_p1'] == 'page_1':
248                 return redirect(url_for('page1'))
249             else:
250                 pass
251 
252             gpio_modif()
253 
254 
255         return render_template('page2.html', button=buttonSts_p2, color=color,
        time=current_time, warning=warning)
256 
257     @app.route("/settings", methods = ['POST', 'GET'])
258     def settings(setting=None):
259         if g.user.id == 3:
260             return redirect(url_for('page1'))
261 
262         if not g.user.username == "admin":
263             return redirect(url_for('login'))
264 
265         current_time = getTime()
266 
267         if request.method == 'POST':
268             check1 = request.form.get('Auto on')
269             time1 = request.form.get('time Auto on')
270             check2 = request.form.get('Auto off')
271             time2 = request.form.get('time Auto off')
272 
273             file =
            open('/home/pi/controle-des-lumieres-knx/5_Programmation/INT_SERV/config.csv',
             "w", newline='')
274             header = ['name','state','param1']
275             csvf = csv.DictWriter(file, fieldnames=header)
276 
277             csvf.writeheader()
278             csvf.writerow({'name': 'Auto on', 'state': check1, 'param1': time1})
279             csvf.writerow({'name': 'Auto off', 'state': check2, 'param1': time2})
280 
281             file.close()
282 
```

```
283             return redirect(url_for('page1'))
284
285         with
            open('/home/pi/controle-des-lumieres-knx/5_Programmation/INT_SERV/config.csv',
            "r") as f:
286             csvreader = csv.reader(f)
287             header = next(csvreader)
288             rows = []
289             for row in csvreader:
290                 rows.append(row)
291
292         f.close()
293
294         return render_template('settings.html', time=current_time, settings=rows,
            warning=warning)
295
296     @app.before_first_request
297     def activate_job():
298         def run_job():
299             logging.debug("starting while loop")
300             while True:
301                 t = getTime()
302
303                 with
                    open('/home/pi/controle-des-lumieres-knx/5_Programmation/INT_SERV/config.c
                    sv', "r") as f:
304                     csvreader = csv.reader(f)
305                     header = next(csvreader)
306                     rows = []
307                     for row in csvreader:
308                         rows.append(row)
309                 f.close()
310
311                 day = datetime.datetime.today().weekday()
312                 #0 monday / 6 sunday
313
314                 #ALL ON and ALL OFF
315                 if rows[0][1] == 'on' and t == rows[0][2] and day < 5:
316                     logging.info("All light ON Automaticly")
317                     buttonSts_p2[0] = "/static/img/img_on.png"
318                     for i in range(8):
319                         buttonSts_p1[i] = "/static/img/img_on.png"
320                         color[i] = "#FFFFFF"
321                     gpio_modif()
322
323                 if rows[1][1] == 'on' and t == rows[1][2] and day < 5:
324                     logging.info("All light ON Automaticly")
325                     buttonSts_p2[1] = "/static/img/img_off.png"
326                     for i in range(8):
327                         buttonSts_p1[i] = "/static/img/img_off.png"
328                         buttonSts_p2[i] = "/static/img/img_off.png"
329                         color[i] = "#333333"
330                     gpio_modif()
331
332                 with open('/sys/class/thermal/thermal_zone0/temp', 'r') as ftemp:
333                     global warning
334                     temp = int(ftemp.read()) / 1000
335                     if temp > 60:
336                         logging.warning("Temp = " + str(int(temp)) + "°")
337                         warning = "Temp = " + str(int(temp)) + "°"
338                     else:
339                         warning = ""
340
341                 time.sleep(60)
342
343         thread = threading.Thread(target=run_job)
344         thread.start()
345
346     if __name__ == "__main__":
347         logging.info("Strating server")
348         app.run(host='0.0.0.0', port=80, debug=False)
349         GPIO.cleanup()
```