

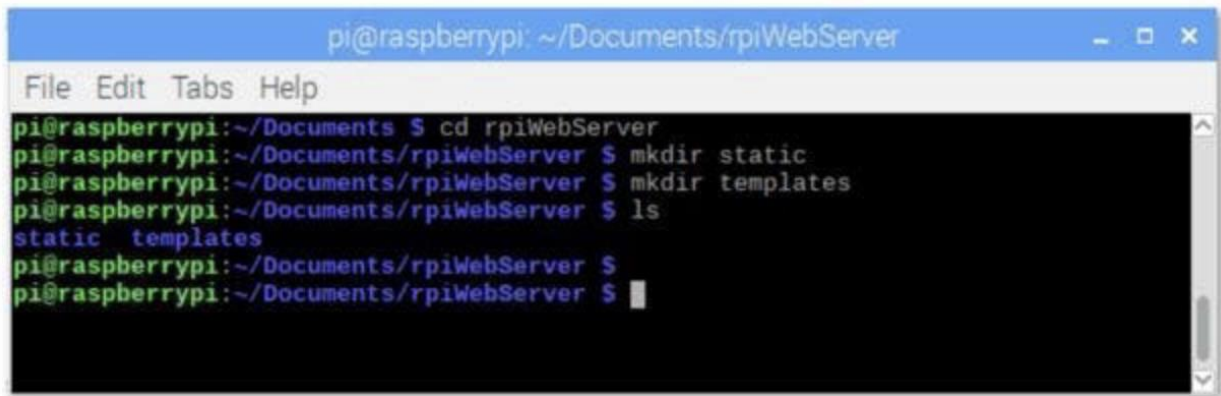
ДОДАТОК (скріншоти програмної реалізації)

```
<!DOCTYPE html>
  <head>
    <title>{{ title }}</title>
  </head>
  <body>
    <h1>Hello, World!</h1>
    <h2>The date and time on the server is: {{ time }}</h2>
  </body>
</html>
```

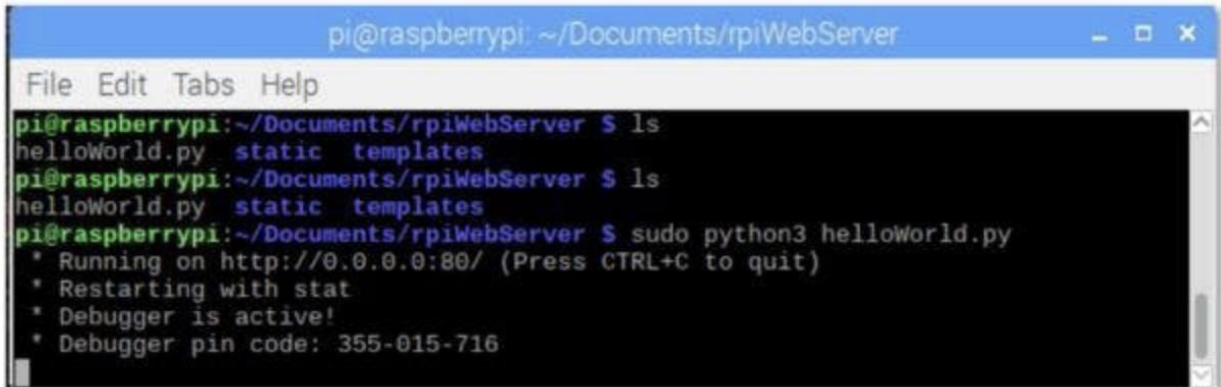
```
'''
Code created by Matt Richardson
for details, visit: http://mattrichardson.com/Raspberry-Pi-Flask/inde...
'''
from flask import Flask, render_template
import datetime
app = Flask(__name__)
@app.route("/")
def hello():
    now = datetime.datetime.now()
    timeString = now.strftime("%Y-%m-%d %H:%M")
    templateData = {
        'title' : 'HELLO!',
        'time': timeString
    }
    return render_template('index.html', **templateData)
if __name__ == "__main__":
    app.run(host='0.0.0.0', port=80, debug=True)
```

```
<!DOCTYPE html>
<head>
  <title>{{ title }}</title>
  <link rel="stylesheet" href="../static/style.css/">
</head>
<body>
  <h1>Hello, World!</h1>
  <h2>The date and time on the server is: {{ time }}</h2>
</body>
</html>
```

```
<!DOCTYPE html>
<head>
  <title>{{ title }}</title>
  <link rel="stylesheet" href="../static/style.css/">
</head>
<body>
  <h1>Hello, World!</h1>
  <h2>The date and time on the server is: {{ time }}</h2>
</body>
</html>
```



```
pi@raspberrypi: ~/Documents/rpiWebServer
File Edit Tabs Help
pi@raspberrypi:~/Documents $ cd rpiWebServer
pi@raspberrypi:~/Documents/rpiWebServer $ mkdir static
pi@raspberrypi:~/Documents/rpiWebServer $ mkdir templates
pi@raspberrypi:~/Documents/rpiWebServer $ ls
static templates
pi@raspberrypi:~/Documents/rpiWebServer $
pi@raspberrypi:~/Documents/rpiWebServer $
```



```
pi@raspberrypi: ~/Documents/rpiWebServer
File Edit Tabs Help
pi@raspberrypi:~/Documents/rpiWebServer $ ls
helloWorld.py static templates
pi@raspberrypi:~/Documents/rpiWebServer $ ls
helloWorld.py static templates
pi@raspberrypi:~/Documents/rpiWebServer $ sudo python3 helloWorld.py
* Running on http://0.0.0.0:80/ (Press CTRL+C to quit)
* Restarting with stat
* Debugger is active!
* Debugger pin code: 355-015-716
```

```

...
        Raspberry Pi GPIO Status and Control
...
import RPi.GPIO as GPIO
from flask import Flask, render_template, request
app = Flask(__name__)
GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)
#define sensors GPIOs
кнопка = 20
senPIR = 16
#define приводи GPIOs
ledRed = 13
ledYlw = 19
ledGrn = 26
#initialize змінні стану GPIO
buttonSts = 0
senPIRSts = 0
ledRedSts = 0
ledYlwSts = 0
ledGrnSts = 0
# і кнопку PIR датчика шпильки Визначити в якості вхідного сигналу
GPIO.setup(button, GPIO.IN)
GPIO.setup(senPIR, GPIO.IN)
# Визначте світлодіодні контакти як вихід
GPIO.setup(ledRed, GPIO.OUT)
GPIO.setup(ledYlw, GPIO.OUT)
GPIO.setup(ledGrn, GPIO.OUT)
# вимкніть світлодіоди
GPIO.output(LEDRed, GPIO.LOW)
GPIO.output(ledYlw, GPIO.LOW)
GPIO.output(ledGrn, GPIO.LOW)

@app.route("/")
def index():
    # Читання
    кнопки стану GPIOSts = GPIO.input(button)
    senPIRSts = GPIO.input(senPIR)
    ledRedSts = GPIO.input (ledRed)
    ledYlwSts = GPIO.input (ledYlw)
    ledGrnSts = GPIO.input (ledGrn)
    templateData = {
        'кнопка': buttonSts,
        'senPIR': senPIRSts,
        'ledRed': ledRedSts,
        'ledYlw': ledYlwSts,
        ' ledGrn': ledGrnSts,
    }
    return render_template('index.html', **templateData)

@app.route("/<deviceName>/<action>")
def action(deviceName, action):
    якщо deviceName == 'ledRed':
        actuator = ledRed,

```