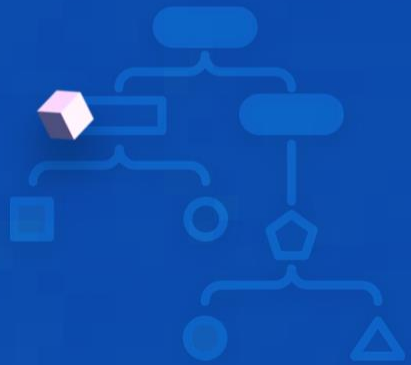
강원혁신플랫폼

리눅스프로그래밍

flask 웹서버, MariaDB 연동



파이썬에서 Django 웹서버는 강력하고 전문적인 웹 프레임워크로, 완전한 기능 세트를 제공하는 반면, 웹 애플리케이션을 빠르고 간편하게 개발할 수 있는 마이크로 웹 프레임워크로 다양한 기능을 간단하게 추가할 수 있으며 HTML 템플릿을 사용하여 동적인 콘텐츠를 생성할 수 있는 마이크로 웹 프레임워크는 무엇인가요?

flask 웹서버

학습목표

학습 내용

1 flask 웹서버, MariaDB 연동

학습 목표

flask 웹서버와 MariaDB를 연동할 수 있다.

flask 웹서버, MariaDB 연동

helloflaskdb.py


```
(.venv) $ cat helloflaskdb.py
from flask import Flask, render_template, json, request
import pymysql.cursors

app = Flask(__name__)
```

helloflaskdb.py


```
# Connect to the database
connection = pymysql.connect(host='localhost',
                             user='scott',
                             password='tiger',
                             database='mydb',
                             cursorclass=pymysql.cursors.DictCursor,
                             charset='utf8'
                             )
```

```
# Route for displaying the sensor data
@app.route('/')
def display_sensor_data():
    with connection.cursor() as cursor:
        # Fetch data from Sensors table
        cursor.execute('SELECT * FROM Sensors')
        sensors_data = cursor.fetchall()
        print(type(sensors_data)) # <class 'list'>
        print(sensors_data)
```

```
[{'id': 1, 'name': 'Sensor1', 'location': 'Location1'}, {'id': 2, 'name': 'Sensor2', 'location': 'Location2'}, {'id': 3, 'name': 'Sensor3', 'location': 'Location3'}]
```

```
return render_template('index.html', sensors=sensors_data,
sensor_data=sensor_data_data, sensor_status=sensor_status_data)

if __name__ == '__main__':
    app.run(debug=True)
```

templates/index.html


```
(.venv) $ cat templates/index.html
<!DOCTYPE html>
<html>
<head>
    <title>Sensor Data</title>
</head>
<body>
    <h1>Sensors</h1>
    <table>
        <thead>
            <tr>
```

```
        <th>ID</th>
        <th>Name</th>
        <th>Location</th>
    </tr>
</thead>

<tbody>
    {% for sensor in sensors %}
        <tr>
```

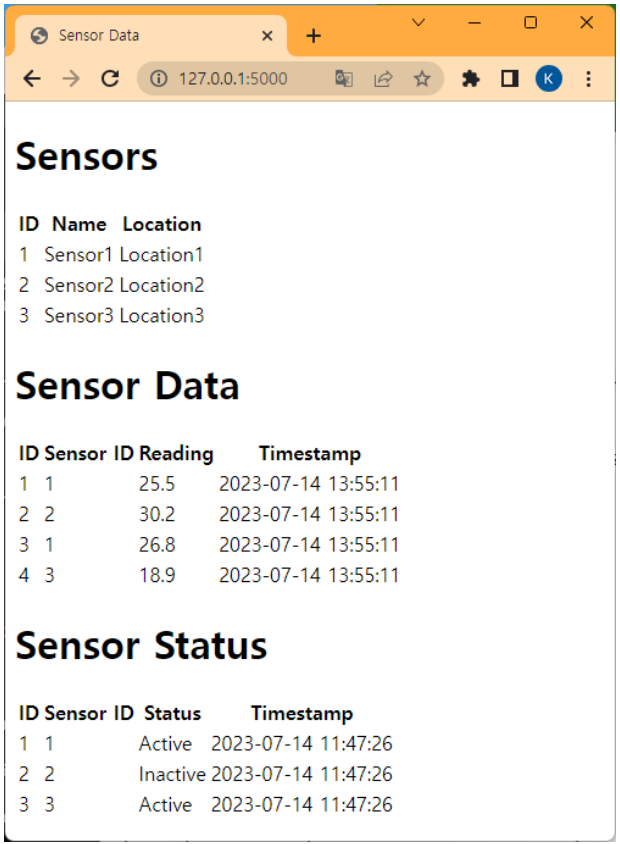
```
            {% for item, value in sensor.items(): %}
                <td>{{ value }}</td>
            {% endfor %}
            </tr>
        {% endfor %}
    </tbody>
</table>
/* 중간 생략 */

</body>
</html>
```

helloflaskdb.py 실행


```
(.venv) $ export FLASK_APP=helloflaskdb
(.venv) $ flask run
 * Serving Flask app 'helloflaskdb'
 * Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment.
Use a production WSGI server instead.
 * Running on http://127.0.0.1:5000
Press CTRL+C to quit
```

helloflaskdb.py 실행



helloflaskdb.py 전체코드


```
from flask import Flask, render_template, json, request
import pymysql.cursors

app = Flask(__name__)

# Connect to the database
connection = pymysql.connect(host='localhost',
                             user='scott',
                             password='tiger',
                             database='mydb',
                             cursorclass=pymysql.cursors.DictCursor,
                             charset='utf8'
```

helloflaskdb.py 전체코드


```
)

# Route for displaying the sensor data
@app.route('/')
def display_sensor_data():
    with connection.cursor() as cursor:
        # Fetch data from Sensors table
        cursor.execute('SELECT * FROM Sensors')
        sensors_data = cursor.fetchall()
        print(type(sensors_data))
        print(sensors_data)
```

helloflaskdb.py 전체코드


```
# Fetch data from SensorData table
  cursor.execute('SELECT * FROM SensorData')
  sensor_data_data = cursor.fetchall()
  print(sensor_data_data)

  # Fetch data from SensorStatus table
  cursor.execute('SELECT * FROM SensorStatus')
  sensor_status_data = cursor.fetchall()
  print(sensor_status_data)
```

helloflaskdb.py 전체코드


```
        # Close the cursor and connection
        #cursor.close()
        #connection.close()

        return render_template('index.html', sensors=sensors_data,
sensor_data=sensor_data_data, sensor_status=sensor_status_data)

if __name__ == '__main__':
    app.run(debug=True)
```

templates/index.html 전체코드


```
(.venv) $ cat templates/index.html
<!DOCTYPE html>
<html>
<head>
    <title>Sensor Data</title>
</head>
<body>
    <h1>Sensors</h1>
    <table>
        <thead>
            <tr>
                <th>ID</th>
```

templates/index.html 전체코드


```
<th>Name</th>
            <th>Location</th>
        </tr>
    </thead>
    <tbody>
        {% for sensor in sensors %}
            <tr>
            {% for item, value in sensor.items(): %}
                <td>{{ value }}</td>
            {% endfor %}
            </tr>
        {% endfor %}
```

templates/index.html 전체코드


```
</tbody>
    </table>

    <h1>Sensor Data</h1>
    <table>
        <thead>
            <tr>
                <th>ID</th>
                <th>Sensor ID</th>
                <th>Reading</th>
                <th>Timestamp</th>
            </tr>
```

templates/index.html 전체코드


```
</thead>
        <tbody>
            {% for data in sensor_data %}
                <tr>
                {% for item, value in data.items(): %}
                    <td>{{ value }}</td>
                {% endfor %}
                </tr>
            {% endfor %}
        </tbody>
    </table>
```

templates/index.html 전체코드


```
<h1>Sensor Status</h1>
    <table>
        <thead>
            <tr>
                <th>ID</th>
                <th>Sensor ID</th>
                <th>Status</th>
                <th>Timestamp</th>
            </tr>
        </thead>
        <tbody>
            {% for status in sensor_status %}
```

templates/index.html 전체코드


```
<tr>
            {% for item, value in status.items(): %}
                <td>{{ value }}</td>
            {% endfor %}
            </tr>
        {% endfor %}
    </tbody>
  </table>
</body>
</html>
```

01 • flask 웹서버, MariaDB 연동