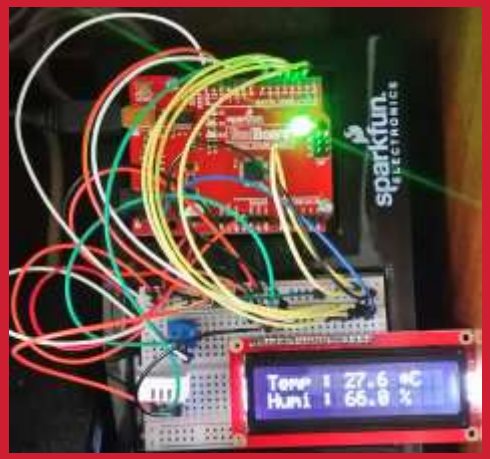
Arduino-IOT

[wk10]



cds_dht22 + node

MongoDB-II

Visualization of Signals using Arduino, Node.js & storing signals in MongoDB & mining iot data using Python

Drone-IoT-Comsi, INJE University

2nd semester, 2022

Email : chaos21c@gmail.com

My ID

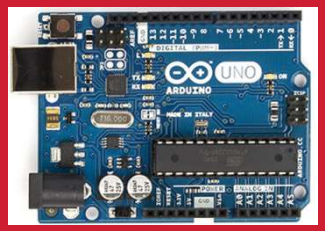
ID를 확인하고 github에 repo 만들기

AA01	강대진	AA13	박제홍
		AA14	심준혁
AA03	김성우	AA15	이상혁
AA04	김정헌	AA16	이승무
		AA17	이승준
AA06	김창연	AA18	이준희
AA07	김창욱	AA19	이현준
AA08	김태화	AA20	임태형
AA09	남승현	AA21	정동현
AA10	류재환		
AA11	박세훈	AA23	정희서
AA12	박신영	AA24	최재형

위의 id를 이용해서 github에 repo를 만드시오.

Option: 아두이노응용 실습 과제 – AAnn

Public, README.md check



[Practice]

- [wk09: cds_dht22]
  - RT Data Visualization with node.js
  - Multiple data and Usage of gauge.js
  - Complete your real-time WEB charts
  - Upload folder: aann-rpt09
  - Use repo “aann” in github

wk09 : Practice : aann-rpt09

- [Target of this week]
  - Complete your works : **cds_dht22 project.**
  - **Save your outcomes and upload outputs in github**

제출폴더명 : **aann-rpt09**

- 제출할 파일들

① **AAnn_cds_dht22_data.png**

② **AAnn_signals_cds_dht22.html**

③ **AAnn_cds_dht22.html**

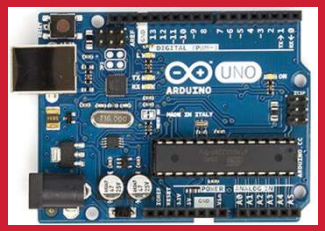
④ **AAnn_cds_dht22.png**

⑤ **All *.ino**

⑥ **All *.js**

⑦ **All *.html**

Email : chaos21c@gmail.com



[Practice]

- [wk09: mongoDB test]
  - Insert documents to test mongodb
  - Upload folder: aann-rpt09
  - Use repo “aann” in github

wk09 : Practice : aann-rpt09

[Target of this week]

- Complete your works : **mongoDB test.**
- **Save your outcomes and upload outputs in github**

제출폴더명 : **aann-rpt09** 에 아래 파일을 추가

- 제출할 파일들

① **AAnn_mongo_schemas.png**

② **AAnn_mongo_update.png**

③ **dbtest.js (cds_dht22 folder)**

④ **dbtest2.js (cds_dht22 folder)**

Email : chaos21c@gmail.com

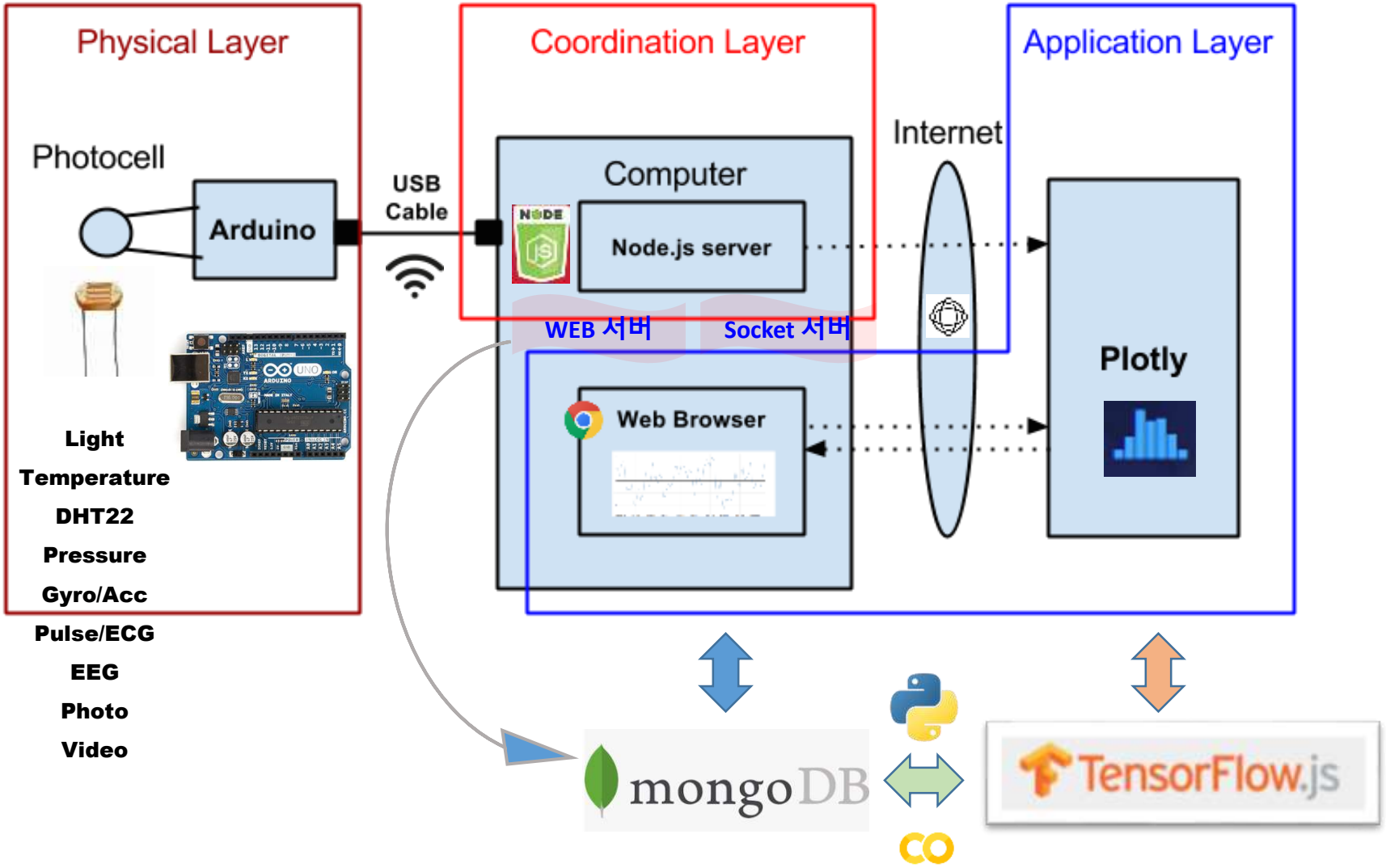
Purpose of AA

주요 수업 목표는 다음과 같다.

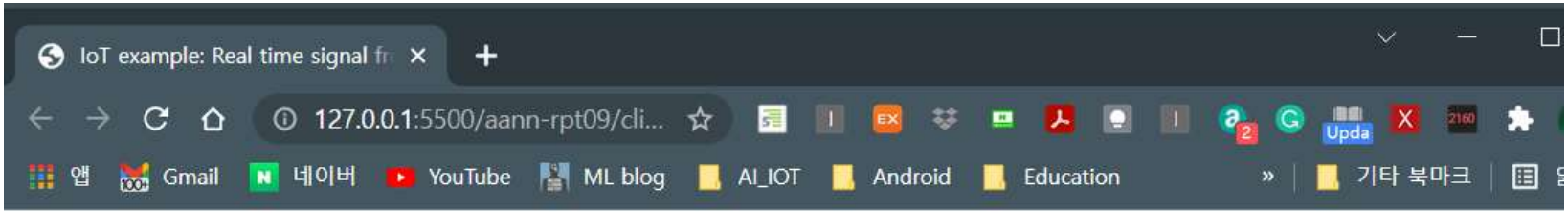
1. Node.js를 이용한 아두이노 센서 신호 처리
2. Plotly.js를 이용한 아두이노 센서 신호 시각화
3. MongoDB에 아두이노 센서 데이터 저장 및 처리

4. 저장된 IoT 데이터의 마이닝 (파이썬 코딩)

Layout [H S C]


Physical Layer
Photocell
Arduino
USB Cable
Light
Temperature
DHT22
Pressure
Gyro/Acc
Pulse/ECG
EEG
Photo
Video
Coordination Layer
Computer
Node.js server
WEB 서버
Socket 서버
Web Browser
Internet
Application Layer
Plotly
mongoDB
TensorFlow.js

on WEB monitoring Arduino data



IoT Signal from Arduino Weather Station

Real-time Signals

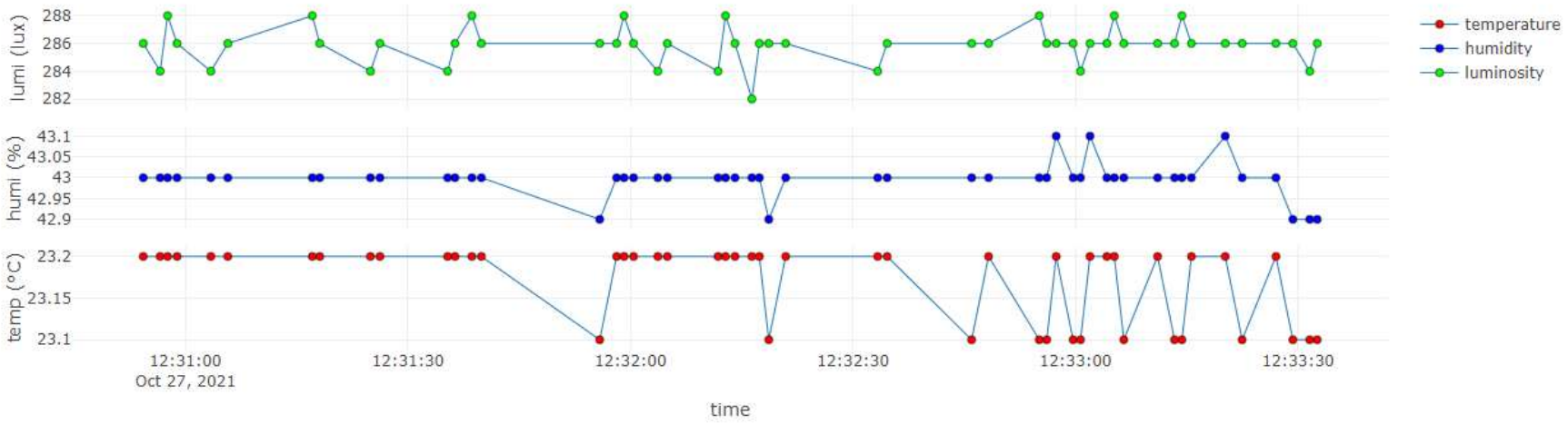
---

on Time: 2021-10-27 11:54:48.997

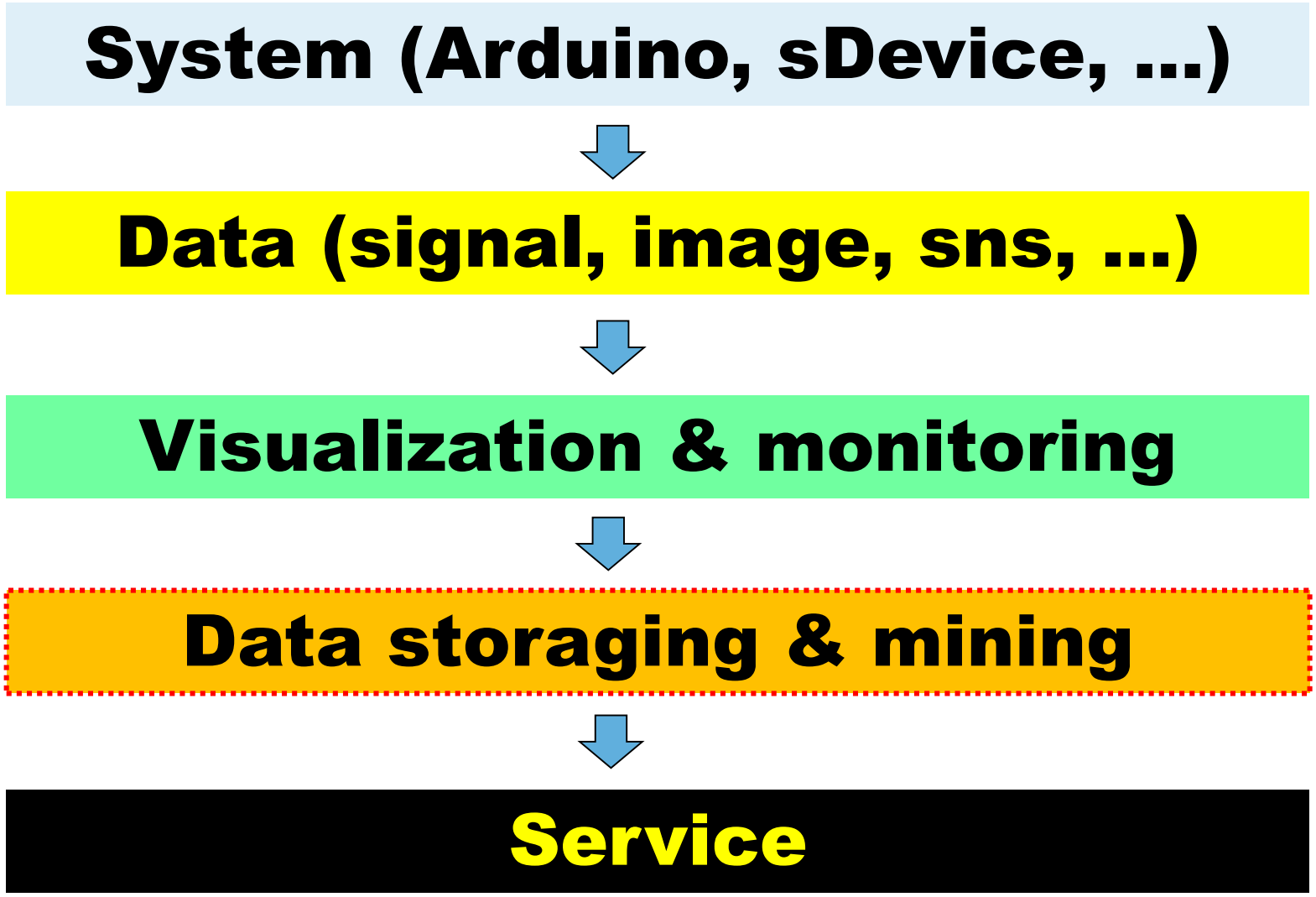
Signals (온도,습도,조도) : 23.4,42.6,286

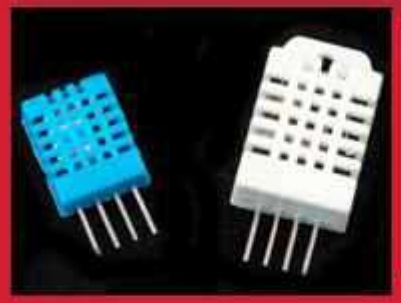
Real-time Weather Station from sensors

on Time: 2021-10-27 12:33:32.600



A5. Introduction to IoT service





[Goal]

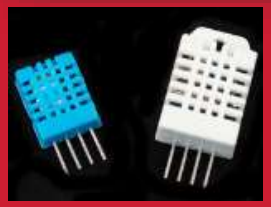
Arduino + Node.js

+ plotly.js

+ MongoDB

→ Data storaging

& visualization

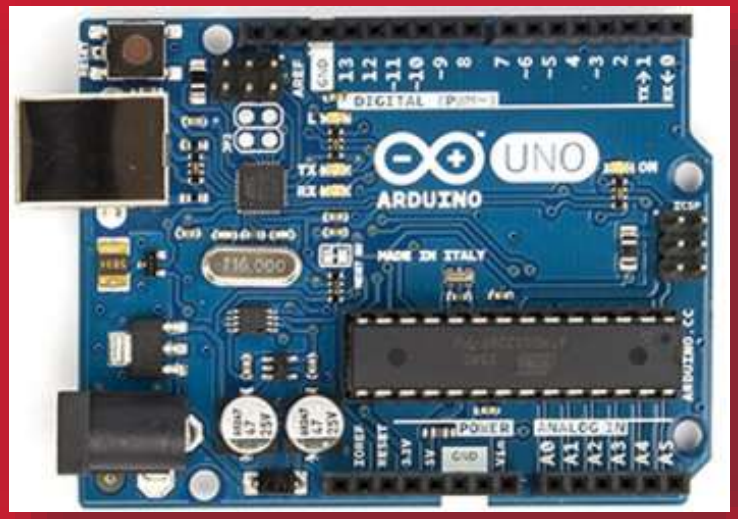


MongoDB from Arduino with node.js & mongoose

```
> show dbs
aa00    0.000GB
admin   0.000GB
config  0.000GB
iot     0.000GB
iot2    0.000GB
iot3    0.001GB
local   0.000GB
test    0.000GB
test2   0.000GB
>
```

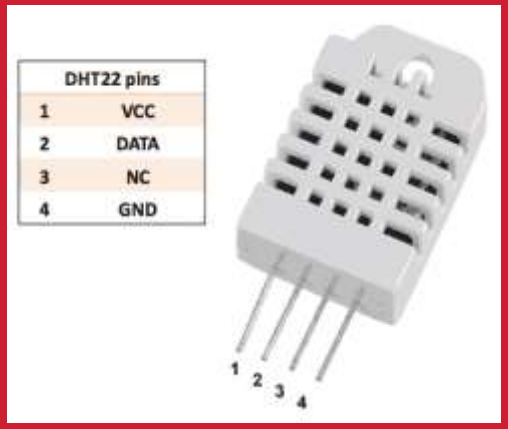
```
mongo db connection OK.
iotInfo: Current date: 2021-11-03 10:52:19.797, Temp: 23.5, Humi: 40.5, Lux: 84
iotInfo: Current date: 2021-11-03 10:52:22.074, Temp: 23.5, Humi: 40.5, Lux: 51
iotInfo: Current date: 2021-11-03 10:52:24.352, Temp: 23.5, Humi: 40.5, Lux: 81
iotInfo: Current date: 2021-11-03 10:52:26.629, Temp: 23.5, Humi: 40.8, Lux: 29
iotInfo: Current date: 2021-11-03 10:52:28.911, Temp: 23.5, Humi: 40.9, Lux: 82
iotInfo: Current date: 2021-11-03 10:52:31.188, Temp: 23.5, Humi: 40.8, Lux: 56
iotInfo: Current date: 2021-11-03 10:52:33.466, Temp: 23.5, Humi: 40.8, Lux: 83
iotInfo: Current date: 2021-11-03 10:52:35.744, Temp: 23.5, Humi: 40.8, Lux: 84
iotInfo: Current date: 2021-11-03 10:52:38.021, Temp: 23.5, Humi: 40.8, Lux: 84
iotInfo: Current date: 2021-11-03 10:52:40.299, Temp: 23.5, Humi: 40.8, Lux: 84
iotInfo: Current date: 2021-11-03 10:52:42.576, Temp: 23.5, Humi: 40.8, Lux: 84
iotInfo: Current date: 2021-11-03 10:52:44.854, Temp: 23.5, Humi: 40.8, Lux: 84
```

Arduino & Node.js & MongoDB

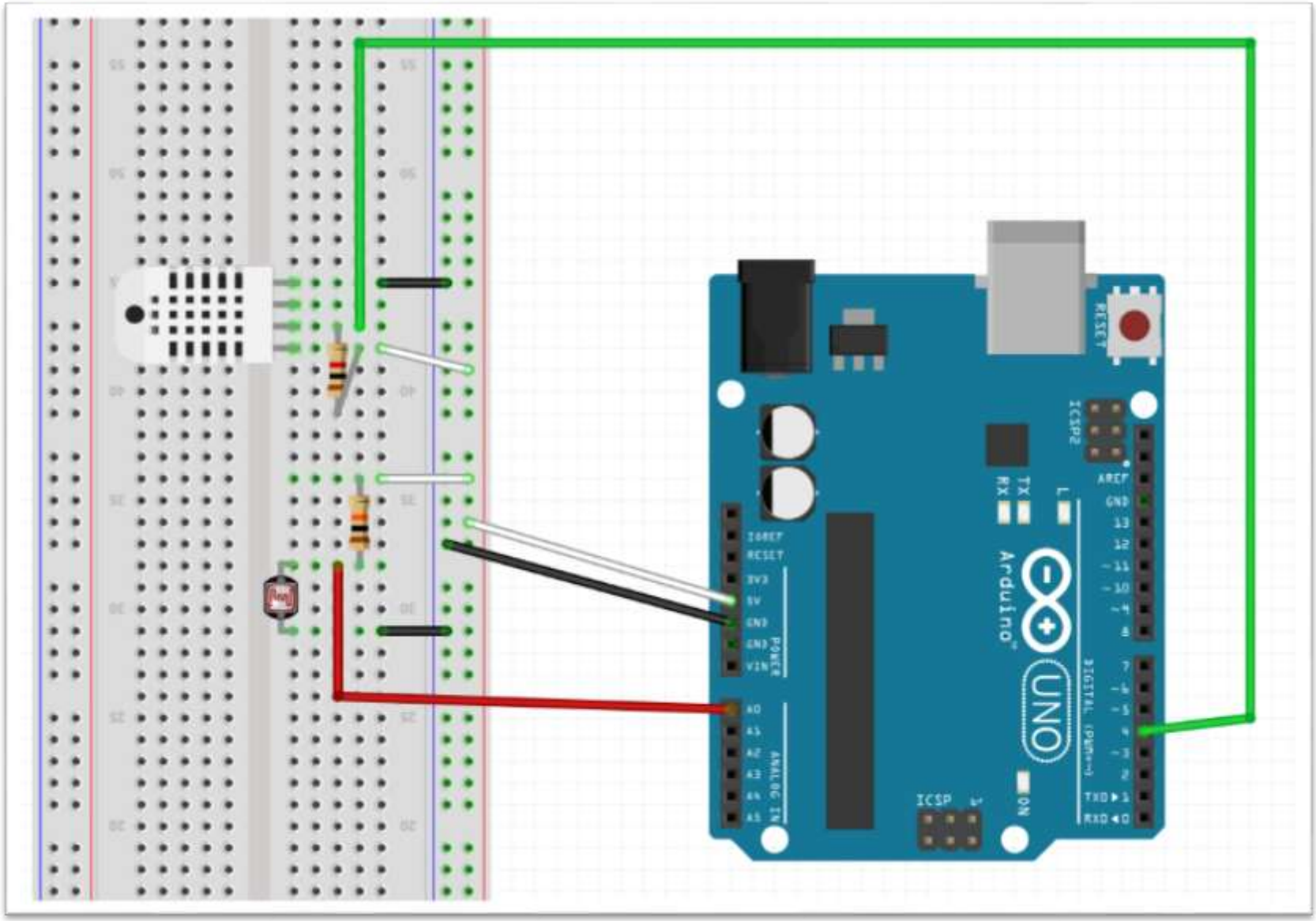


Multi-sensors

DHT22 + CdS



DHT22 + CdS circuit



DHT22[D4] + 1 kΩ, CdS[A0] + 10 k Ω

DHT22 + CdS + Node.js

[3] Result: Parsed streaming data from dht22 & CdS (Run in Terminal)

COM3

```
21.0,24.7,205
21.0,24.7,207
21.0,24.7,205
21.0,24.7,152
21.0,24.7,167
20.9,24.6,166
20.9,24.6,204
21.0,24.8,204
21.0,24.8,152
21.0,24.8,173
21.0,24.8,191
21.0,24.8,203
21.0,24.8,207
21.0,24.9,204
21.0,24.9,204
```

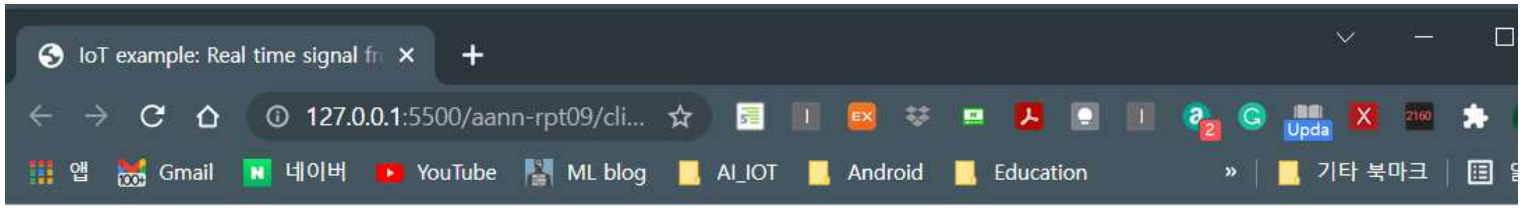
☑ 자동 스크롤 ☐ 타임스탬프

node cds_dht22_node.js

문제 출력 디버그 콘솔 터미널 JUPYTER node

```
AAnn,2021-10-27 11:53:01.872,23.4,42.6,286
AAnn,2021-10-27 11:53:02.872,23.4,42.6,286
AAnn,2021-10-27 11:53:04.150,23.4,42.6,286
AAnn,2021-10-27 11:53:05.154,23.4,42.6,286
AAnn,2021-10-27 11:53:06.428,23.4,42.6,286
AAnn,2021-10-27 11:53:07.431,23.4,42.6,286
AAnn,2021-10-27 11:53:08.709,23.4,42.6,286
AAnn,2021-10-27 11:53:09.713,23.4,42.6,286
AAnn,2021-10-27 11:53:10.987,23.4,42.6,286
AAnn,2021-10-27 11:53:11.990,23.4,42.6,286
AAnn,2021-10-27 11:53:13.269,23.4,42.6,284
AAnn,2021-10-27 11:53:14.268,23.4,42.6,284
AAnn,2021-10-27 11:53:15.546,23.4,42.6,286
AAnn,2021-10-27 11:53:16.550,23.4,42.6,284
AAnn,2021-10-27 11:53:17.824,23.4,42.6,286
AAnn,2021-10-27 11:53:18.827,23.4,42.6,286
```

Arduino data on network socket



IoT Signal from Arduino Weather Station

Real-time Signals

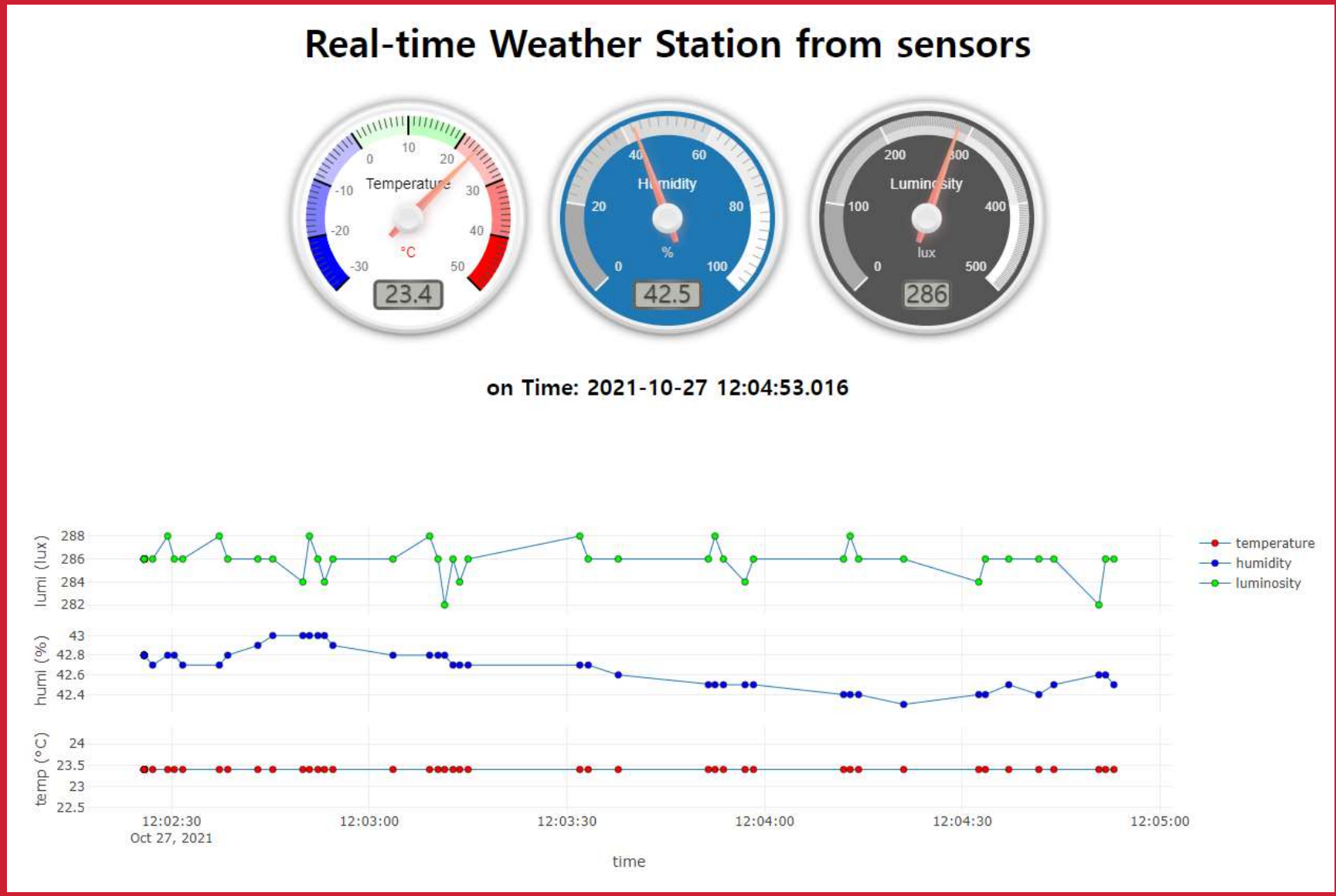
---

on Time: 2021-10-27 11:54:48.997

Signals (온도,습도,조도) : 23.4,42.6,286

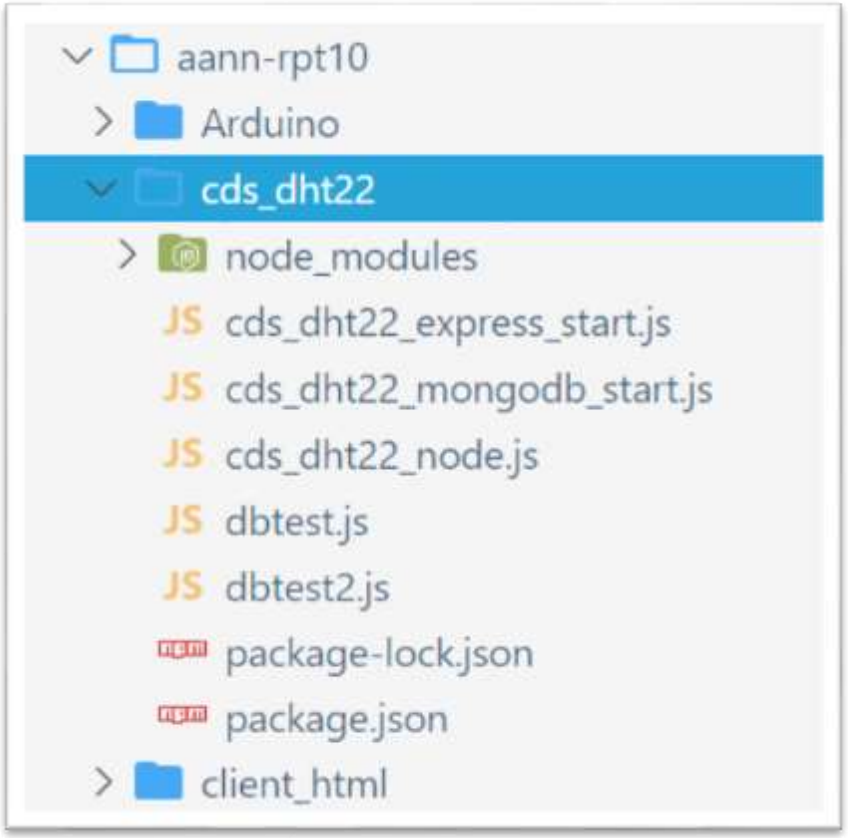
Real-time monitoring of signals from Arduino CdS + DHT22 circuit

WEB client： client_cds_dht22.html



A5.9.5 DHT22 + CdS + Node.js + MongoDB

1. 작업 폴더 구조 [2021]



A5.9.5 DHT22 + CdS + Node.js + MongoDB

2.1 cds_dht22_mongodb.js

```
1  // cds_dht22_mongodb.js
2
3  var serialport = require('serialport');
4  var portName = 'COM4';  // check your COM port!!
5  var port     =    process.env.PORT || 3000;
6
7  var io = require('socket.io').listen(port);
8
9  // MongoDB
10 var mongoose = require('mongoose');
11 var Schema = mongoose.Schema;
12 // MongoDB connection
13 mongoose.connect('mongodb://localhost:27017/iot'); // DB name
14 var db = mongoose.connection;
15 db.on('error', console.error.bind(console, 'connection error:'));
16 db.once('open', function callback () {
17     console.log("mongo db connection OK.");
18 });
19 // Schema
20 var iotSchema = new Schema({
21     date : String,
22     temperature : String,
23     humidity : String,
24     luminosity: String
25 });
```

A5.9.5 DHT22 + CdS + Node.js + MongoDB

2.2 cds_dht22_mongodb.js

```
23  // Schema
24  var iotSchema = new Schema({
25    date: String,
26    temperature: String,
27    humidity: String,
28    luminosity: String,
29  });
30  // Display data on console in the case of saving data.
31  iotSchema.methods.info = function () {
32    var iotInfo = this.date
33      ? "Current date: " +
34        this.date +
35        ", Temp: " +
36        this.temperature +
37        ", Humi: " +
38        this.humidity +
39        ", Lux: " +
40        this.luminosity
41      : "I don't have a date";
42    console.log("iotInfo: " + iotInfo);
43  };
```

A5.9.5 DHT22 + CdS + Node.js + MongoDB

2.3 cds_dht22_mongodb.js

```
45  const Readline = require("@serialport/parser-readline");
46  // serial port object
47  var sp = new serialport(portName, {
48    baudRate: 9600, // 9600  38400
49    dataBits: 8,
50    parity: "none",
51    stopBits: 1,
52    flowControl: false,
53    parser: new Readline("\r\n"),
54  });
55
56  const parser = sp.pipe(new Readline({ delimiter: "\r\n" }));
57
58  // Read the port data
59  sp.on("open", () => {
60    console.log("serial port open");
61  });
62
63  var readData = ""; // this stores the buffer
64  var temp = "";
65  var humi = "";
66  var lux = "";
67  var mdata = []; // this array stores date and data from multiple sensors
68  var firstcommaidx = 0;
69
70  var Sensor = mongoose.model("Sensor", iotSchema); // sensor data model
```

A5.9.5 DHT22 + CdS + Node.js + MongoDB

2.4 cds_dht22_mongodb.js – parsing data & save data in MongoDB

```
72  parser.on("data", function (data) {
73    // call back when data is received
74    readData = data.toString(); // append data to buffer
75    firstcommaidx = readData.indexOf(",");
76    // parsing data into signals
77    if (readData.lastIndexOf(",") > firstcommaidx && firstcommaidx > 0) {
78      temp = readData.substring(0, firstcommaidx);
79      humi = readData.substring(
80        firstcommaidx + 1,
81        readData.indexOf(",", firstcommaidx + 1)
82      );
83      lux = readData.substring(readData.lastIndexOf(",") + 1);
84      readData = "";
85      dStr = getDateString();
86      mdata[0] = dStr; // Date
87      mdata[1] = temp; // temperature data
88      mdata[2] = humi; // humidity data
89      mdata[3] = lux; // luminosity data
90      var iot = new Sensor({
91        date: dStr,
92        temperature: temp,
93        humidity: humi,
94        luminosity: lux,
95      });
96      // save iot data to MongoDB
97      iot.save(function (err, iot) {
98        if (err) return handleEvent(err);
99        iot.info(); // Display the information of iot data  on console.
100     });
101     io.sockets.emit("message", mdata); // send data to all clients
102   } else {
103     // error
104     console.log(readData);
105   }
106 });
```

// Sensor document 객체

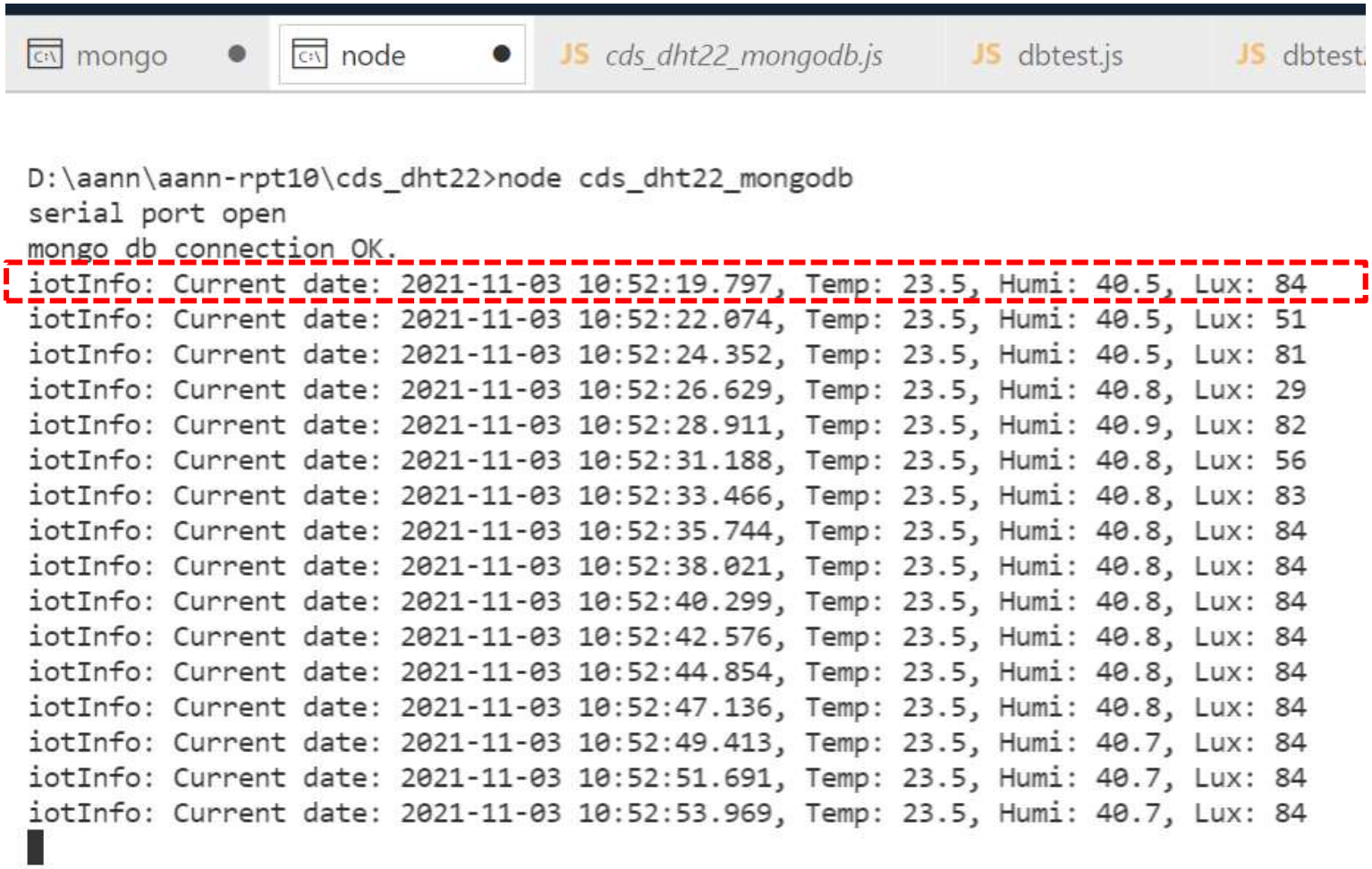
A5.9.5 DHT22 + CdS + Node.js + MongoDB

2.5 cds_dht22_mongodb.js

```
113  io.sockets.on("connection", function (socket) {
114    // If socket.io receives message from the client browser then
115    // this call back will be executed.
116    socket.on("message", function (msg) {
117      console.log(msg);
118    });
119    // If a web browser disconnects from Socket.IO then this callback is called.
120    socket.on("disconnect", function () {
121      console.log("disconnected");
122    });
123  });
124
125  // helper function to get a nicely formatted date string
126  function getDateString() {
127    var time = new Date().getTime();
128    // 32400000 is (GMT+9 Korea, GimHae)
129    // for your timezone just multiply +/-GMT by 3600000
130    var datestr = new Date(time + 32400000)
131      .toISOString()
132      .replace(/T/, " ")
133      .replace(/Z/, "");
134    return datestr;
135  }
```

A5.9.5 DHT22 + CdS + Node.js + MongoDB

2.6 [Run] node cds_dht22_mongodb.js



A5.9.5 DHT22 + CdS + Node.js + MongoDB

3. cds_dht22_mongodb.js → Check documents in Mongo shell

Mongo shell

> show dbs

> use iot

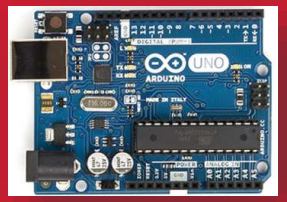
> show collections

> db.sensors.find()
.pretty()

mongo | node | cds_dht22_mongodb.js

```
> show dbs
admin   0.000GB
config  0.000GB
iot     0.000GB
local   0.000GB
test    0.000GB
> use iot
switched to db iot
> show collections
sensors
> db.sensors.find().pretty()
{
        "_id" : ObjectId("6181eb5338cdbc755b232170"),
        "date" : "2021-11-03 10:52:19.797",
        "temperature" : "23.5",
        "humidity" : "40.5",
        "luminosity" : "84",
        "__v" : 0
}
{
        "_id" : ObjectId("6181eb5638cdbc755b232172"),
        "date" : "2021-11-03 10:52:22.074",
        "temperature" : "23.5",
        "humidity" : "40.5",
        "luminosity" : "51",
        "__v" : 0
}
{
        "_id" : ObjectId("6181eb5838cdbc755b232174"),
        "date" : "2021-11-03 10:52:24.352",
        "temperature" : "23.5",
        "humidity" : "40.5",
        "luminosity" : "81",
        "__v" : 0
}
```

Save as
AAnn_iot_mongdb.png



Arduino & Node.js & MongoDB & Express server

mongoDB

node js express

A5.9.6 DHT22 + CdS + Node.js + MongoDB

1.1 Install express server

- Go to cds_dht22 project
- npm install --save express
- package.json

```
D:\aann\aann-rpt10\cds_dht22>npm install --save express
npm WARN cds_dht22@1.0.0 No repository field.

+ express@4.17.1
added 50 packages from 33 contributors, removed 67 packages,
66s

4 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities


D:\aann\aann-rpt10\cds_dht22>
```

1.2 Install express server – package.json

- Go to cds_dht22 project
- npm install --save express
- package.json

```
"author": "aa00",
"license": "MIT",
"dependencies": {
  "express": "^4.17.1",
  "mongoose": "^6.0.12",
  "serialport": "^9.2.4",
  "socket.io": "^2.4.1"
}
```

^6.7.0

A5.9.6 DHT22 + CdS + Node.js + MongoDB

2.1 cds_dht22_express.js

```
1  // cds_dht22_express.js
2  var express = require("express");
3  var app = express();
4  var web_port = 3030; // express port
5
6  // MongoDB
7  var mongoose = require("mongoose");
8  var Schema = mongoose.Schema; // Schema object
9  // MongoDB connection
10 mongoose.connect("mongodb://localhost:27017/iot", {
11   useNewUrlParser: true,
12   useUnifiedTopology: true,
13 });
14 var db = mongoose.connection;
15 db.on("error", console.error.bind(console, "connection error:"));
16 db.once("open", function callback() {
17   console.log("mongo db connection OK.");
18 });
19 // Schema
20 var iotSchema = new Schema({
21   date: String,
22   temperature: String,
23   humidity: String,
24   luminosity: String,
25 });
26 var Sensor = mongoose.model("Sensor", iotSchema); // sensor data model
```

A5.9.6 DHT22 + CdS + Node.js + MongoDB

2.2 cds_dht22_express.js

```
28  // Web routing address
29  app.get("/", function (req, res) {
30    // localhost:3030/
31    res.send("Hello Arduino IOT: express server by AA00!");
32  });
33  // find all data & return them
34  app.get("/iot", function (req, res) {
35    Sensor.find(function (err, data) {
36      res.json(data);
37    });
38  });
39  // find data by id
40  app.get("/iot/:id", function (req, res) {
41    Sensor.findById(req.params.id, function (err, data) {
42      res.json(data);
43    });
44  });
45
46  // Express WEB
47  app.use(express.static(__dirname + "/public")); // WEB root folder
48  app.listen(web_port); // port 3030
49  console.log("Express_IOT is running at port:3030");
```

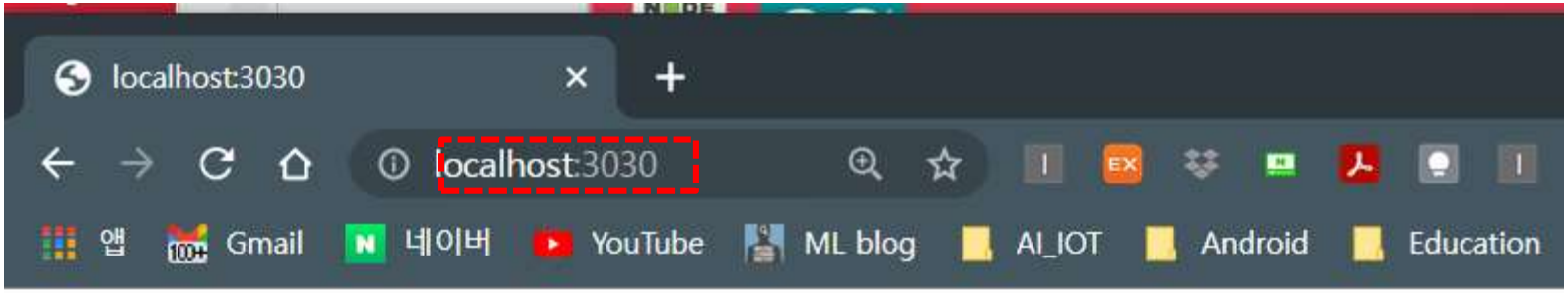
A5.9.6 DHT22 + CdS + Node.js + MongoDB

2.3 [Run] node cds_dht22_express.js

```
(base) D:\aann\aann-rpt10\cds_dht22>node cds_dht22_express
Express_IOT is running at port:3030
mongo db connection OK.
```

A5.9.6 DHT22 + CdS + Node.js + MongoDB

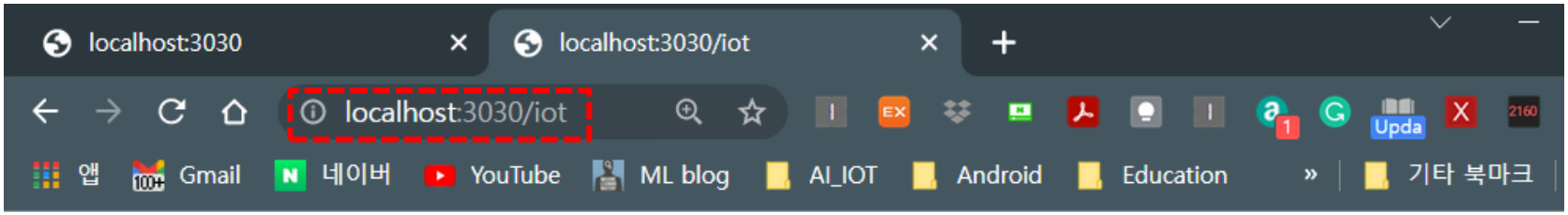
2.4 cds_dht22_express.js → routing1, http://localhost:3030/



Hello Arduino IOT: express server by AA00!

A5.9.6 DHT22 + CdS + Node.js + MongoDB

2.5 cds_dht22_express.js → routing2 http://localhost:3030/iot



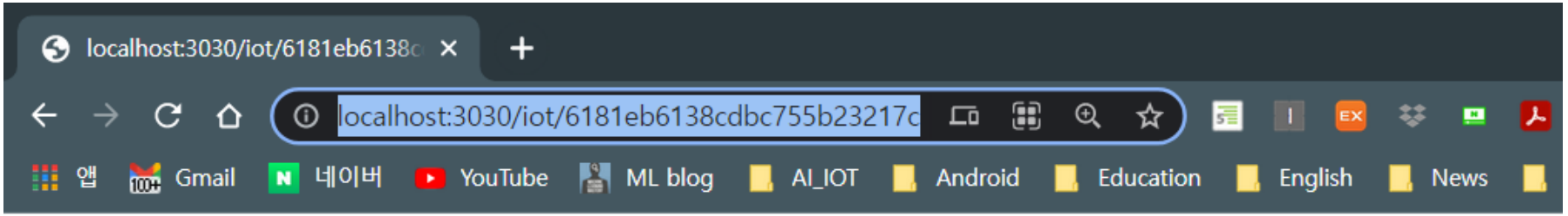

```
[{"_id":"6181eb5338cdbc755b232170","date":"2021-11-03
10:52:19.797","temperature":"23.5","humidity":"40.5","luminosity":"84","__v":0},
{"_id":"6181eb5638cdbc755b232172","date":"2021-11-03
10:52:22.074","temperature":"23.5","humidity":"40.5","luminosity":"51","__v":0},
{"_id":"6181eb5838cdbc755b232174","date":"2021-11-03
10:52:24.352","temperature":"23.5","humidity":"40.5","luminosity":"81","__v":0},
{"_id":"6181eb5a38cdbc755b232176","date":"2021-11-03
10:52:26.629","temperature":"23.5","humidity":"40.8","luminosity":"29","__v":0},
{"_id":"6181eb5c38cdbc755b232178","date":"2021-11-03
10:52:28.911","temperature":"23.5","humidity":"40.9","luminosity":"82","__v":0},
{"_id":"6181eb5f38cdbc755b23217a","date":"2021-11-03
10:52:31.188","temperature":"23.5","humidity":"40.8","luminosity":"56","__v":0},
{"_id":"6181eb6138cdbc755b23217c","date":"2021-11-03
10:52:33.466","temperature":"23.5","humidity":"40.8","luminosity":"83","__v":0},
{"_id":"6181eb6338cdbc755b23217e","date":"2021-11-03
10:52:35.744","temperature":"23.5","humidity":"40.8","luminosity":"84","__v":0},
{"_id":"6181eb6638cdbc755b232180","date":"2021-11-03
10:52:38.021","temperature":"23.5","humidity":"40.8","luminosity":"84","__v":0}
{"_id":"6181eb6838cdbc755b23218
```


Save as

AAnn_iot_mongodb_web.png

A5.9.6 DHT22 + CdS + Node.js + MongoDB

2.6 cds_dht22_express.js → routing2 http://localhost:3030/iot:id

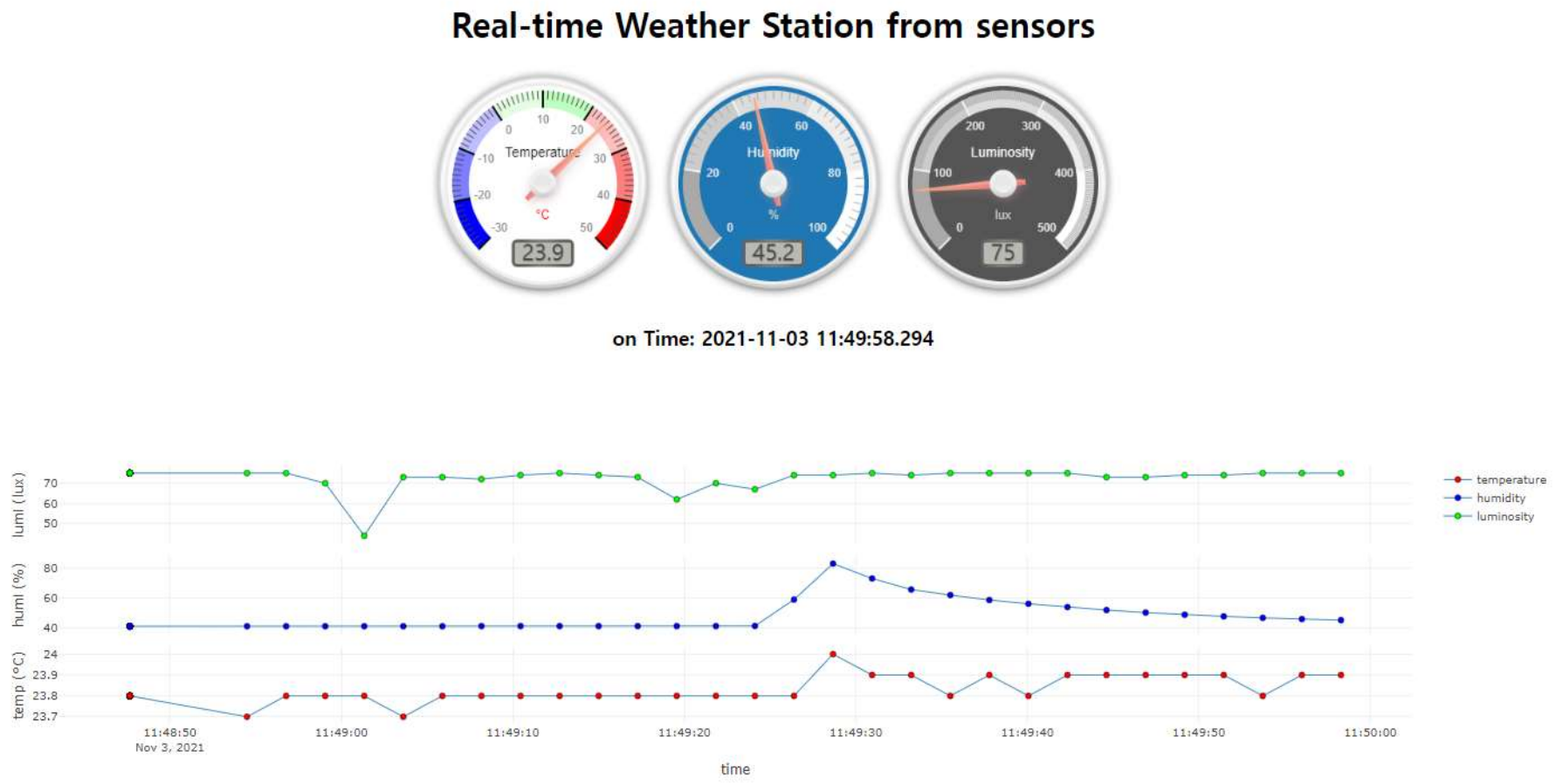


{"_id":"6181eb6138cdbc755b23217c","date":"2021-11-03
10:52:33.466","temperature":"23.5","humidity":"40.8","luminosity":"83","__v":0}

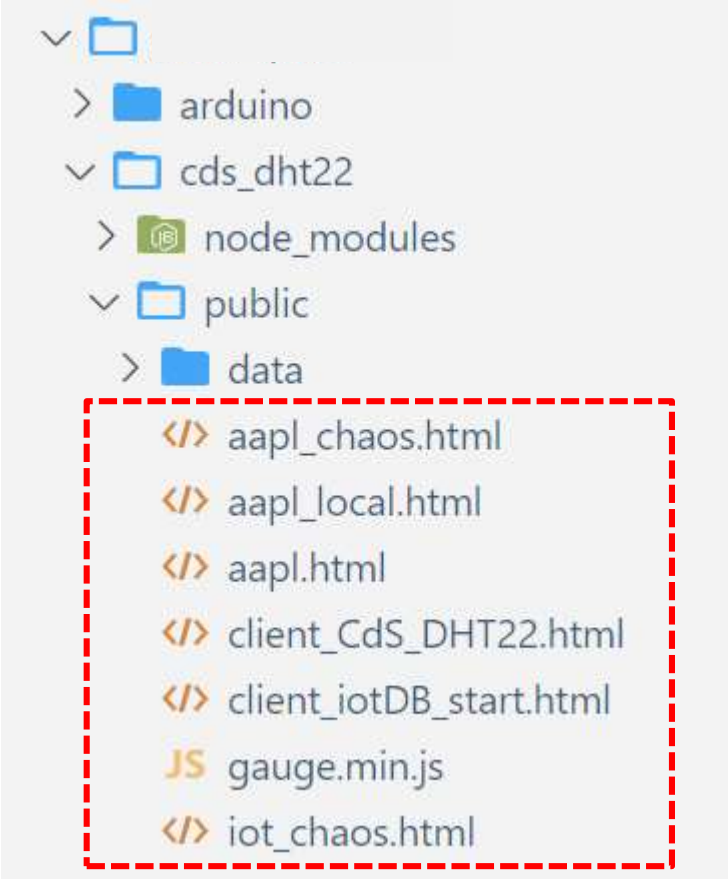
A5.9.6 DHT22 + CdS + Node.js + MongoDB

2.7 copy cds_dht22_client.html & gauge.min.js → ./public/ subfolder

http://localhost:3030/client_cds_dht22.html (web root folder)



2.8 CORS bug (Cross Origin Resource Sharing)



*** CORS problem**

→ 원격 서버 내의 파일에 접근을 허용

A5.9.6 DHT22 + CdS + Node.js + MongoDB

2.9 CORS patch on the express server → cds_dht22_express.js
Node cmd에서 'cors' module 설치 (version 2.8.4 이상)
npm install --save cors

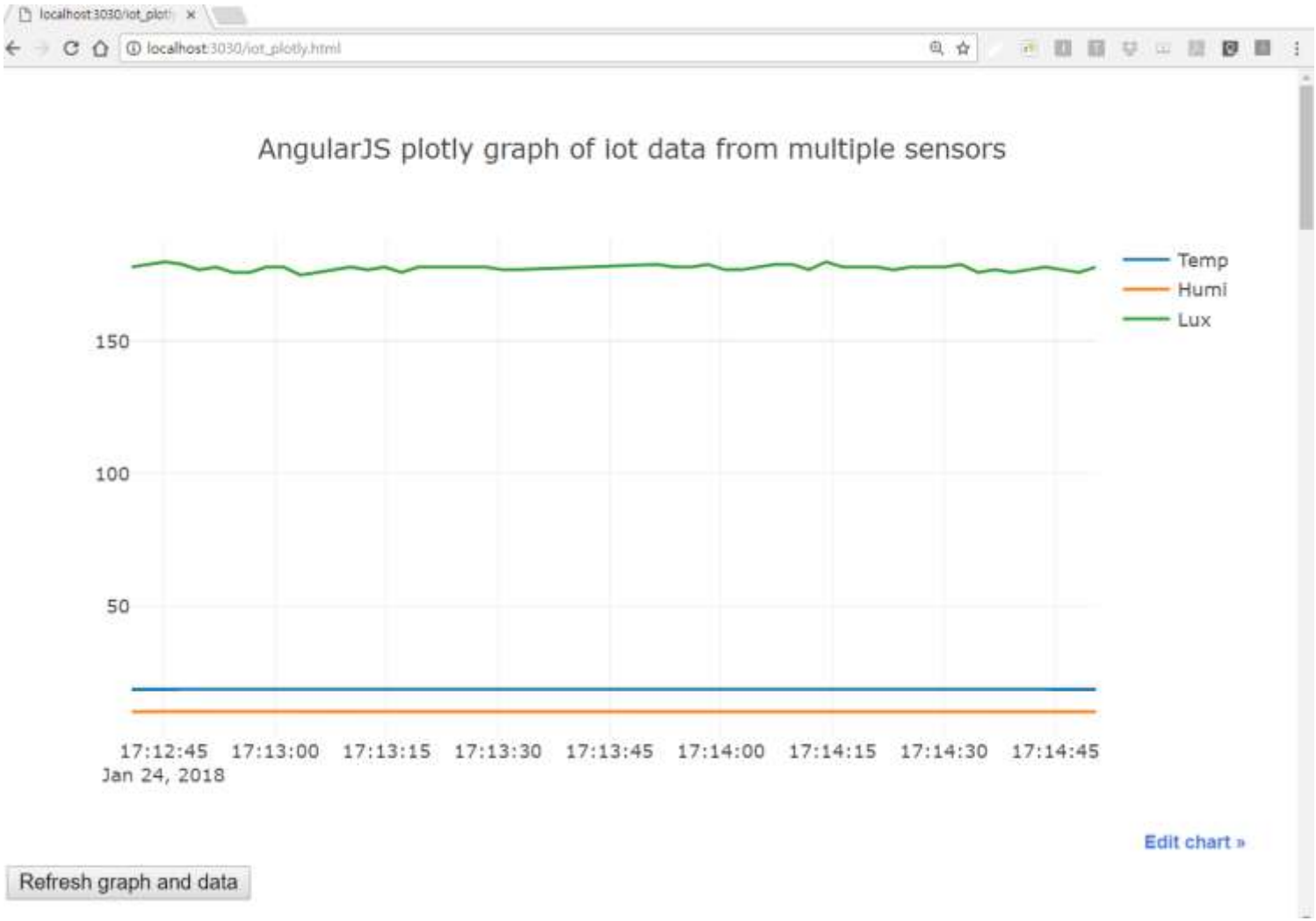
```
1   // cds_dht22_express_cors.js
2   // Express + CORS
3   var express = require("express");
4   var cors = require("cors");
5   var app = express();
6   app.use(cors());
7   var web_port = 3030; // express port
8
9   // MongoDB
10  var mongoose = require("mongoose");
11  var Schema = mongoose.Schema; // Schema object
```

```
D:\aann\aann-rpt11\cds_dht22>node cds_dht22_express
Express_IOT with CORS is running at port:3030
mongo db connection OK.
```

Web monitoring

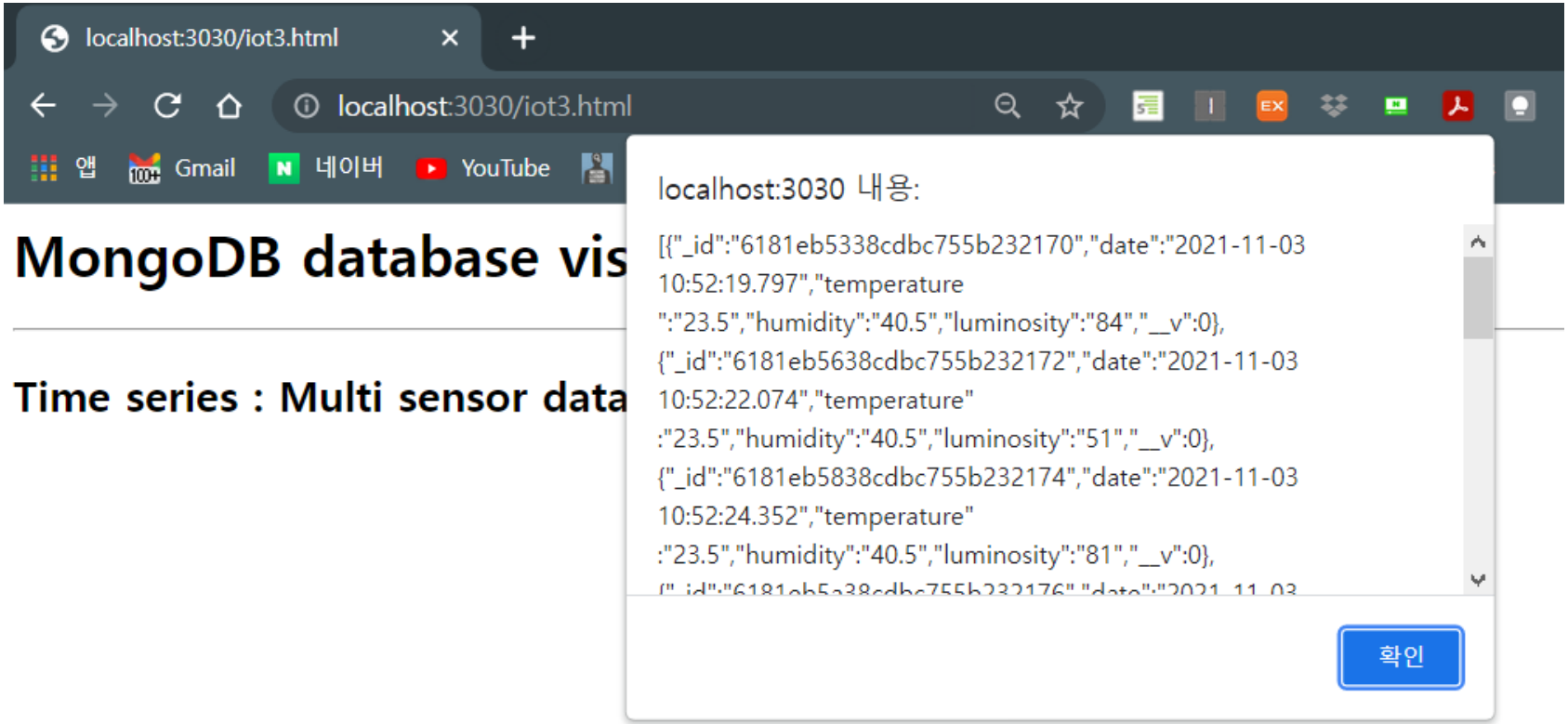
DHT22 + CdS + Node.js + MongoDB

Web monitoring – Google AngularJS



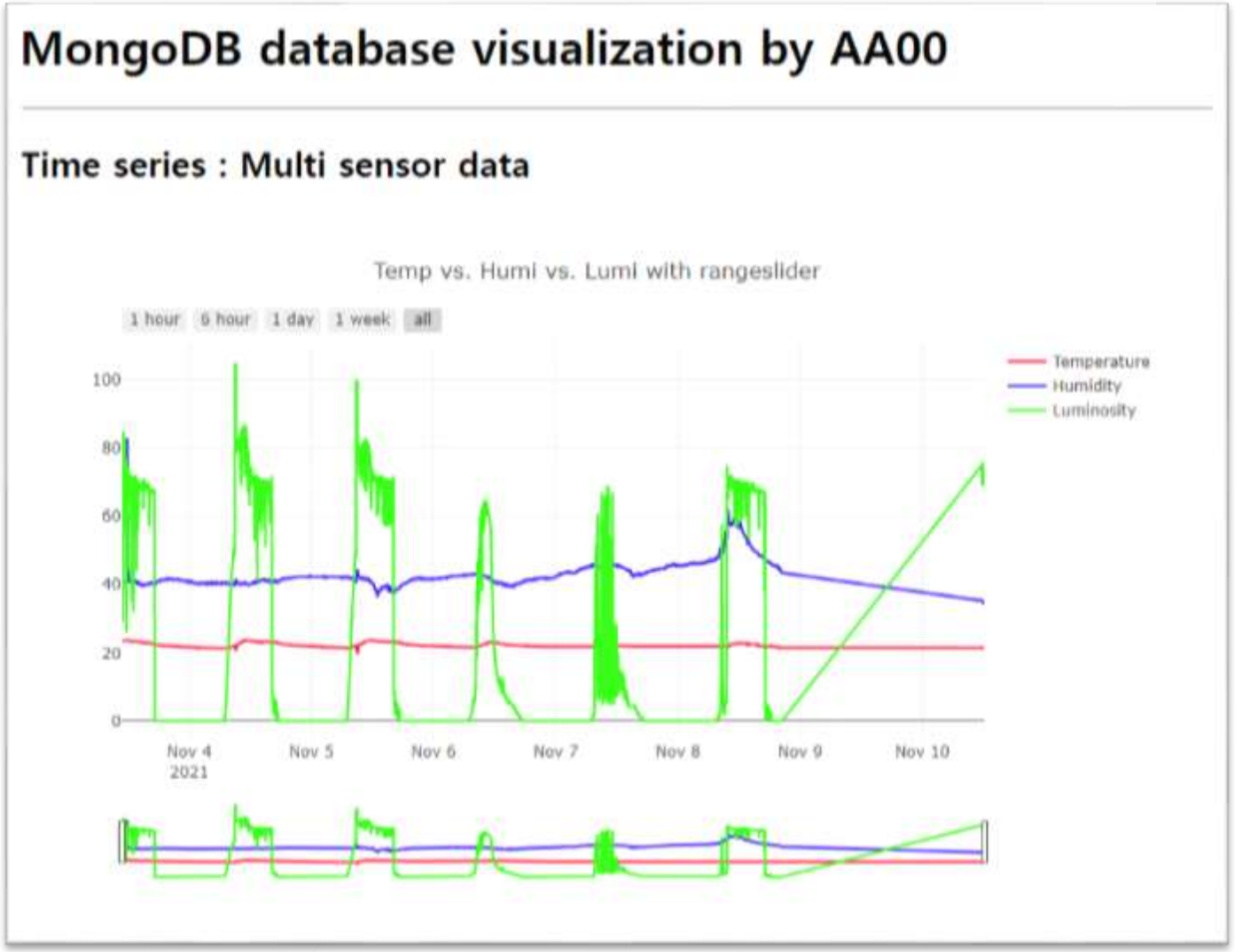
DHT22 + CdS + Node.js + MongoDB

Web monitoring: http://localhost:3030/iot



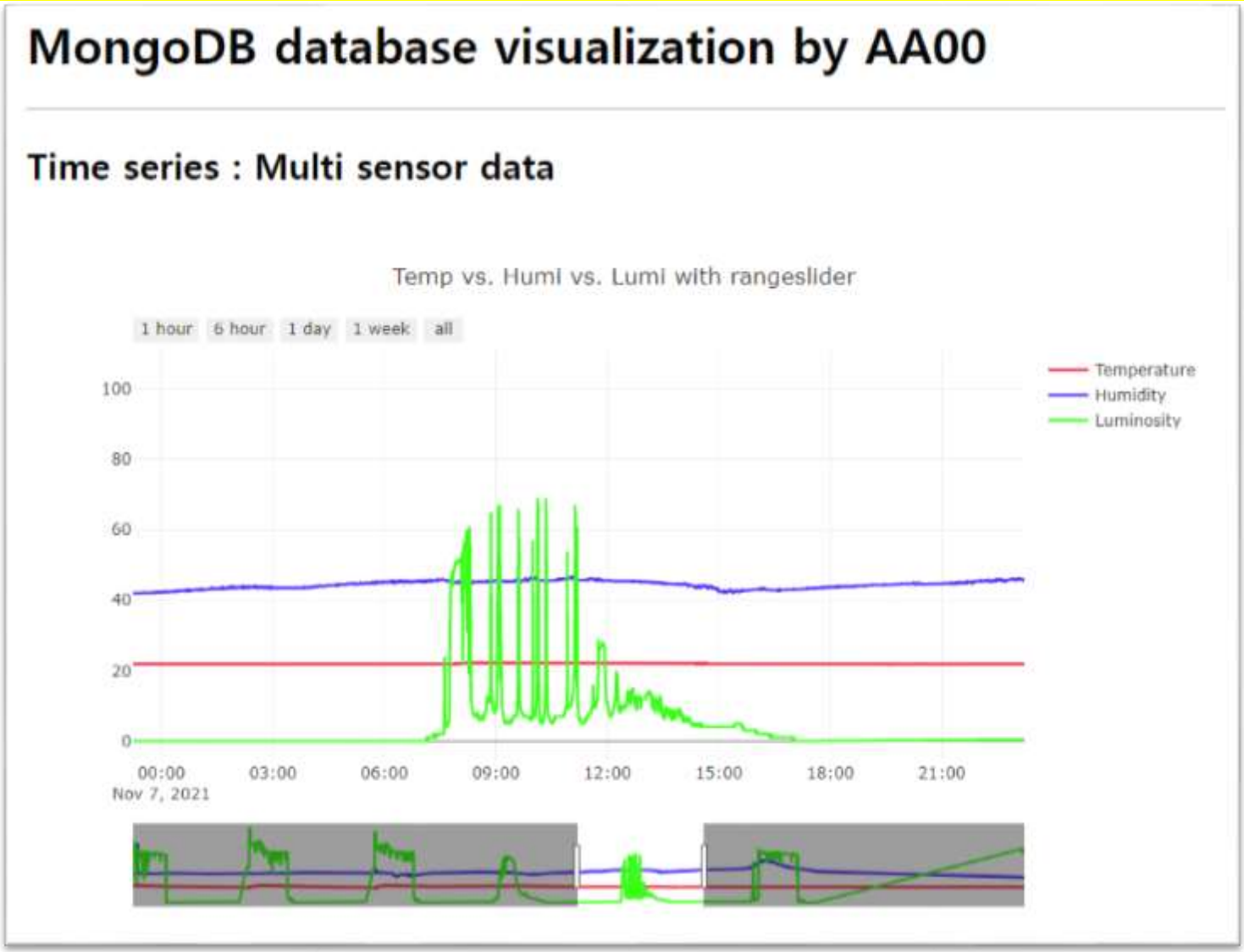
DHT22 + CdS + Node.js + MongoDB

Web monitoring



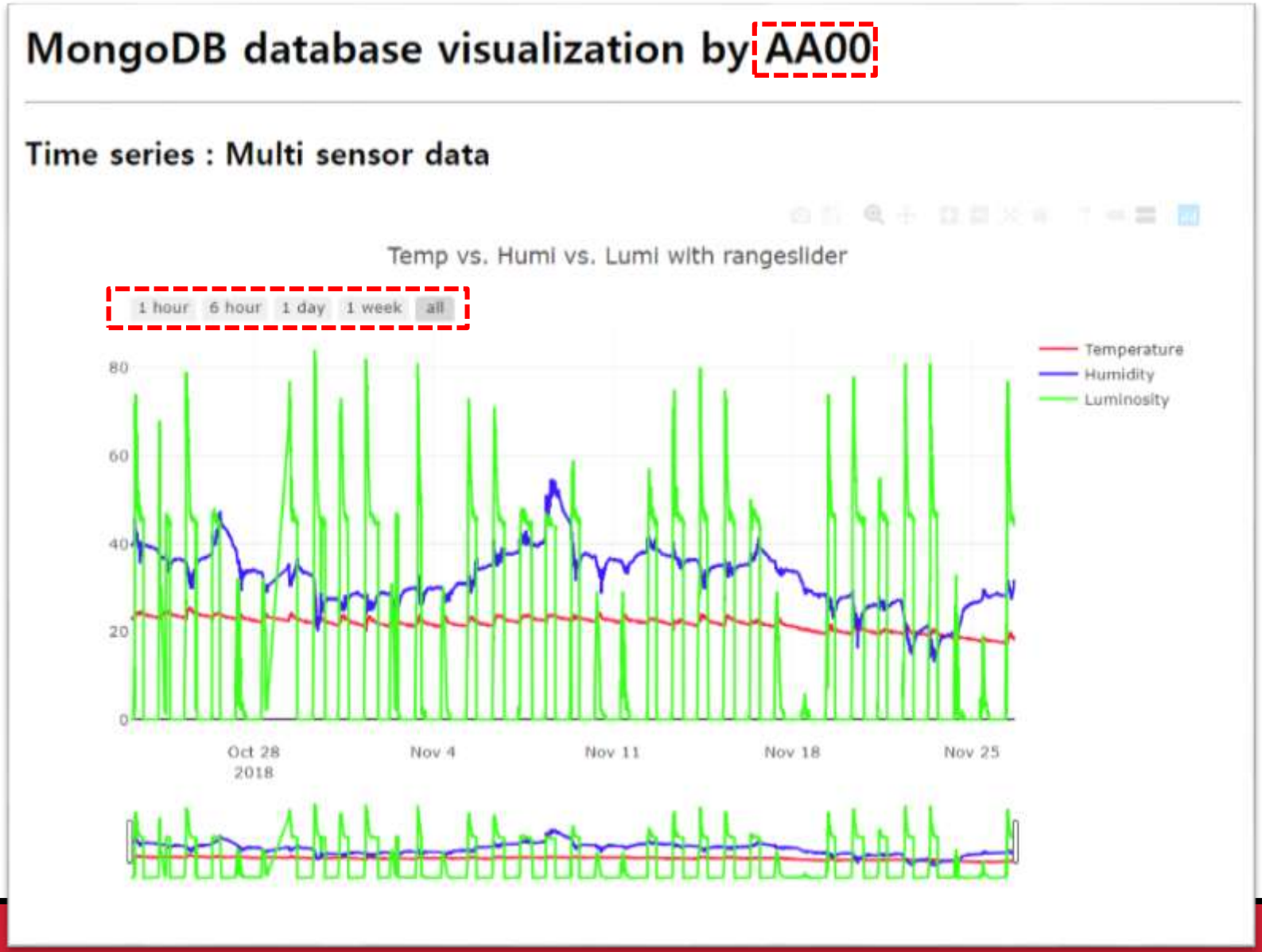
DHT22 + CdS + Node.js + MongoDB

Web monitoring



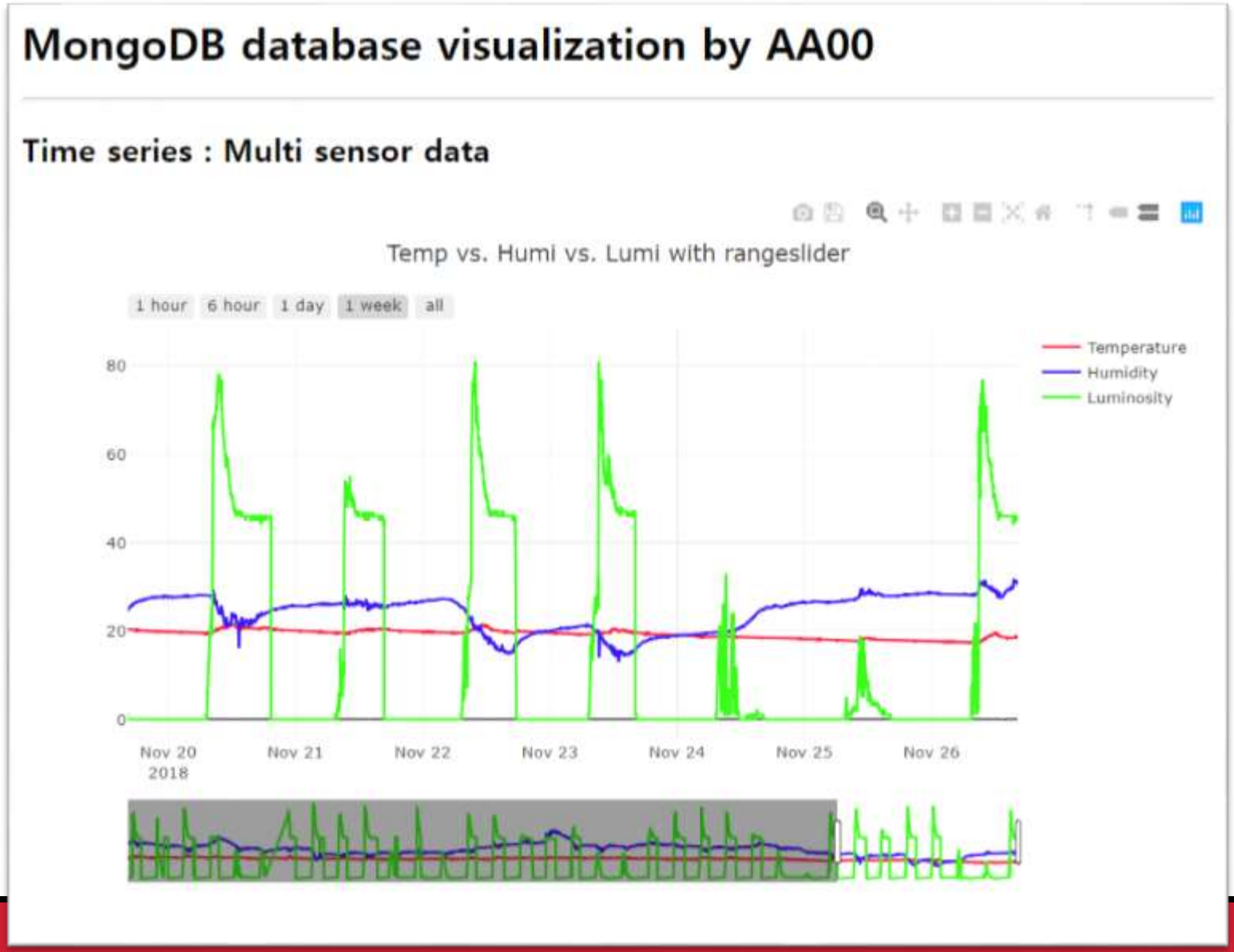
A5.9.7 DHT22 + CdS + Node.js + MongoDB

3.5 Web client: client_iotDB.html – iot DB monitoring (public 폴더에서 제공)



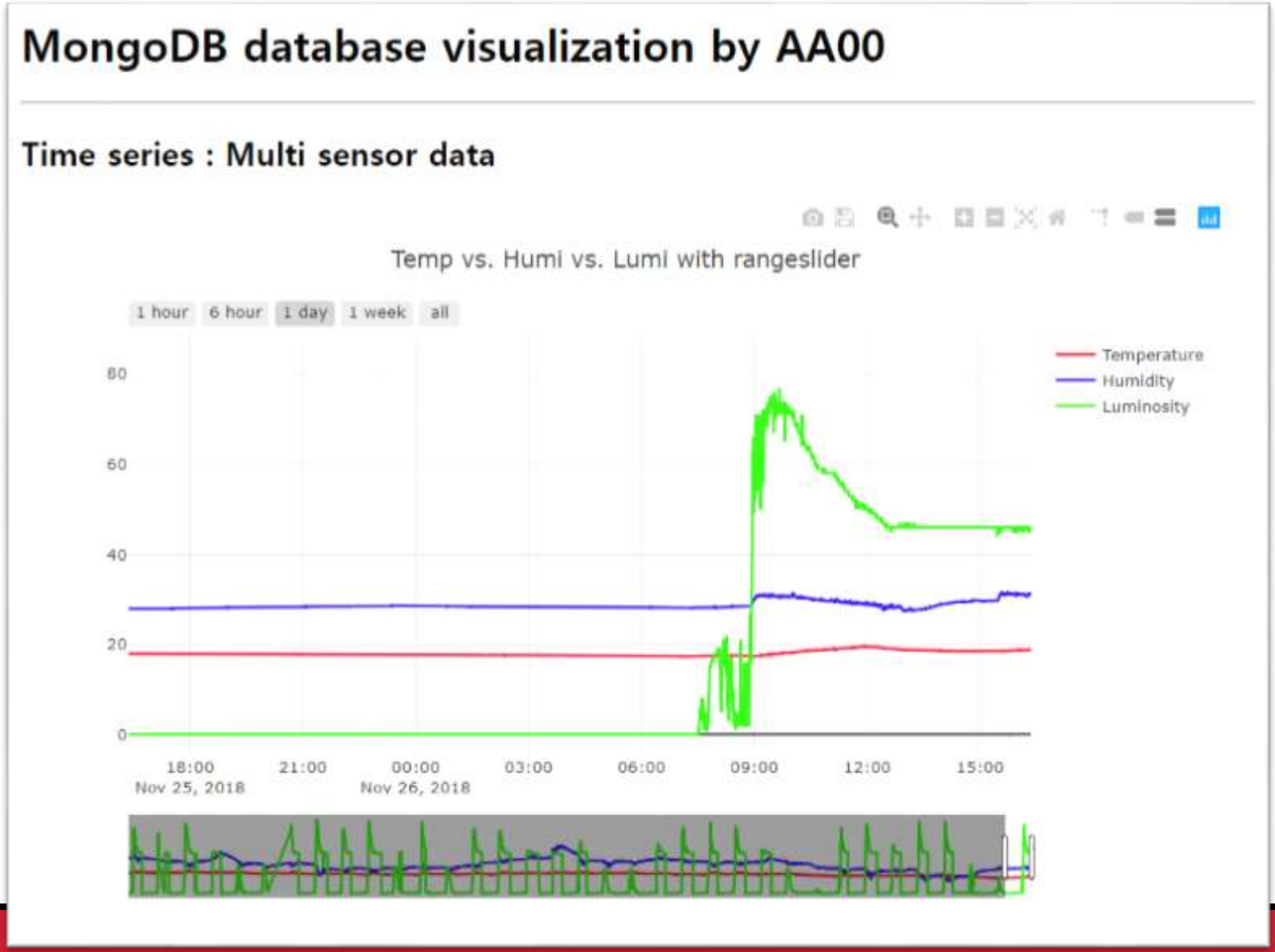
A5.9.7 DHT22 + CdS + Node.js + MongoDB

Web monitoring-2: week



A5.9.7 DHT22 + CdS + Node.js + MongoDB

Web monitoring-3: day



A5.9.8 DHT22 + CdS + Node.js + MongoDB

3.1 Web client: client_iotDB.html

client_iotDB.html

```
1  <!DOCTYPE html>
2  <head>
3      <meta charset="utf-8">
4      <!-- Plotly.js -->
5      <script src="https://cdn.plot.ly/plotly-latest.min.js"></script>
6  </head>
7  <body>
8      <h1>MongoDB database visualization by AA00</h1>
9      <hr>
10     <h2>Time series : Multi sensor data</h2>
11
12     <!-- Plotly chart will be drawn inside this DIV -->
13     <div id="myDiv" style="width: 900px;height: 600px"></div>
14
```

A5.9.7 DHT22 + CdS + Node.js + MongoDB

3.2 Web client: client_iotDB.html

```
<script>
    <!-- JAVASCRIPT CODE GOES HERE -->

    Plotly.d3.json("http://localhost:3030/iot", function(err, json){
        //alert(json);
         alert(JSON.stringify(json));  // It works!!!
        //alert(JSON.parse(eval(json));
        if(err) throw err;

        var date = [];
        var temp = [];
        var humi = [];
        var lumi = [];
        var jsonData = eval(JSON.stringify(json));
        //alert(jsonData.length);
        //alert(jsonData[2].luminosity);

        for (var i = 0; i < jsonData.length; i++) {
            date[i] = jsonData[i].date;
            temp[i] = jsonData[i].temperature ;
            humi[i] = jsonData[i].humidity;
            lumi[i] = jsonData[i].luminosity;

        }
```

JSON file

{"_id":"5fbdab71d02de805786af43c","date":"2020-11-25 09:55:13.068","temperature":"18.9","humidity":"24.7","luminosity":"207","__v":0},
{"_id":"5fbdab73d02de805786af43d","date":"2020-11-25 09:55:15.341","temperature":"18.9","humidity":"24.7","luminosity":"208","__v":0},
{"_id":"5fbdab75d02de805786af43e","date":"2020-11-25

A5.9.7 DHT22 + CdS + Node.js + MongoDB

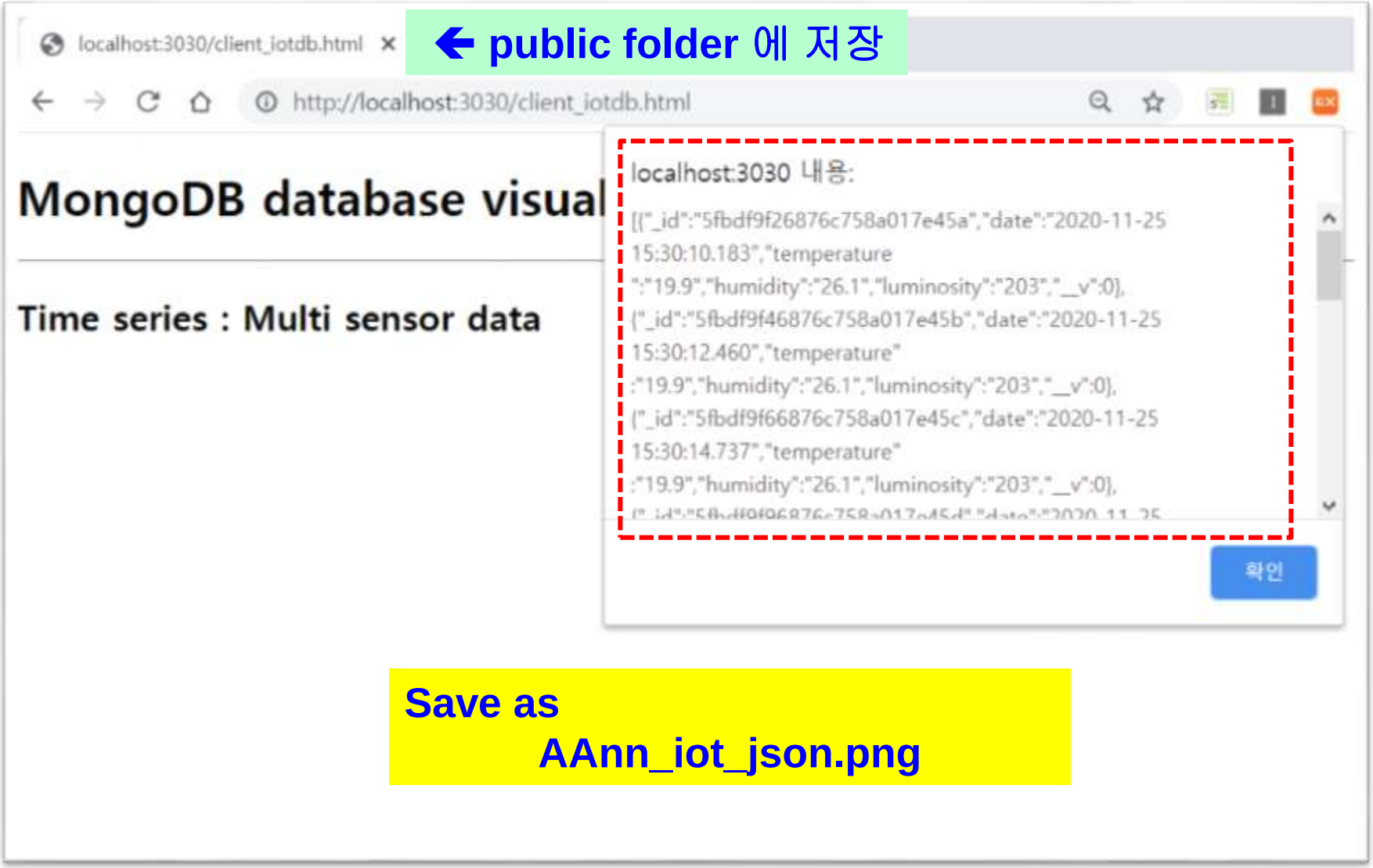
3.3 Web client: client_iotDB.html – data & layout

```
// time series of sensor data
var trace1 = {
    type: "scatter",
    mode: "lines",
    name: 'Temperature',
    x: date,
    y: temp,
    line: {color: '#fc1234'}
}

var trace2 = {
    type: "scatter",
    mode: "lines",
    name: 'Humidity',
    x: date,
    y: humi,
    line: {color: '#3412fc'}
}
var trace3 = {
    type: "scatter",
    mode: "lines",
    name: 'Luminosity',
    x: date,
    y: lumi,
    line: {color: '#34fc12'}
}
var data = [trace1, trace2, trace3];
```

```
// Layout with builtin rangeslider
var layout = {
    title: 'Temp vs. Humi vs. Lumi with rangeslider',
    xaxis: {
        autorange: true,
        range: [date[0], date[date.length-1]],
        rangeselector: {buttons: [
            {
                count: 1,
                label: '1 hour',
                step: 'hour',
                stepmode: 'backward'
            },
            {
                count: 6,
                label: '6 hour',
                step: 'hour',
                stepmode: 'backward'
            },
            {
                count: 24,
                label: '1 day',
                step: 'hour',
                stepmode: 'backward'
            },
            {
                count: 7,
                label: '1 week',
                step: 'day',
                stepmode: 'backward'
            },
            {step: 'all'}
            ]},
        rangeslider: {range: [date[0], date[date.length-1]]},
        type: 'date'
    },
    yaxis: {
        autorange: true,
        range: [0, 300],
        type: 'linear'
    }
};

Plotly.newPlot('myDiv', data, layout);
})
```

A5.9.7 DHT22 + CdS + Node.js + MongoDB

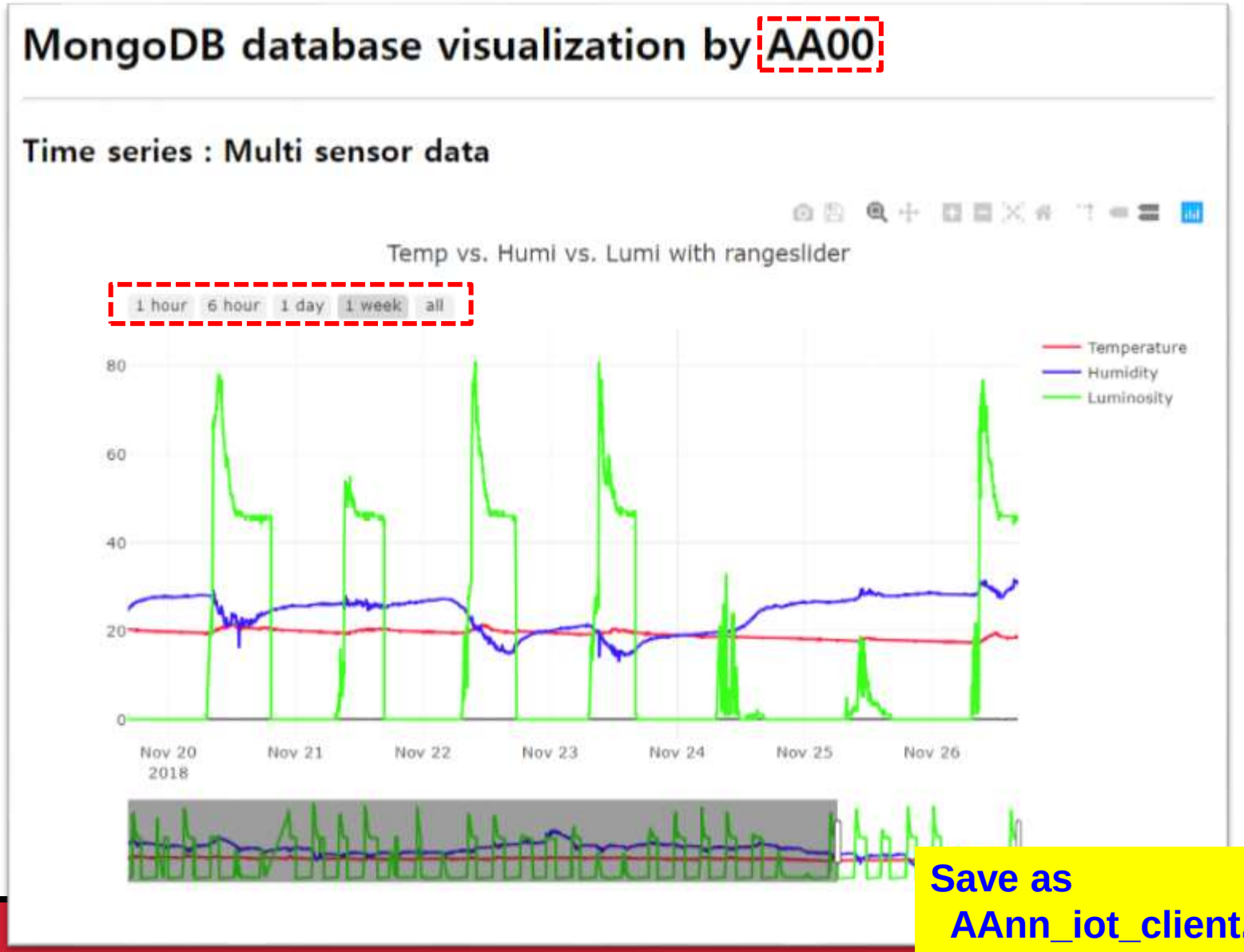
3.4 Web client: client_iotDB.html – load iot data in json file



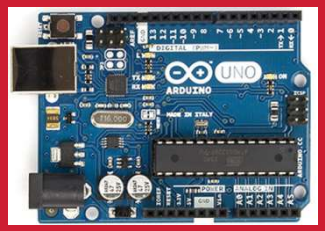
Save as
AAnn_iot_json.png

A5.9.7 DHT22 + CdS + Node.js + MongoDB

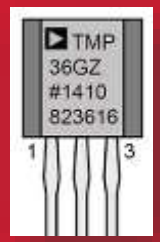
3.5 Web client: client_iotDB.html – iot DB monitoring

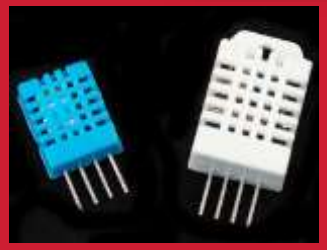


Save as
AAnn_iot_client.png



[Practice]





- [wk10]
  - RT Data storaging with MongoDB
  - Multi-sensor circuits (cds-dht22)
  - Complete your project
  - Upload folder: aann-rpt10

wk10 : Practice : aann-rpt10

- [Target of this week]
  - Complete your works
  - Save your outcomes and upload outputs in github

제출폴더명 : **aann-rpt10**

- 압축할 파일들

① AAnn_iot_mongodb.png

② AAnn_iot_mongodb_web.png

③ AAnn_iot_json.png

④ AAnn_iot_client.png

⑤ All *.ino

⑥ All *.js

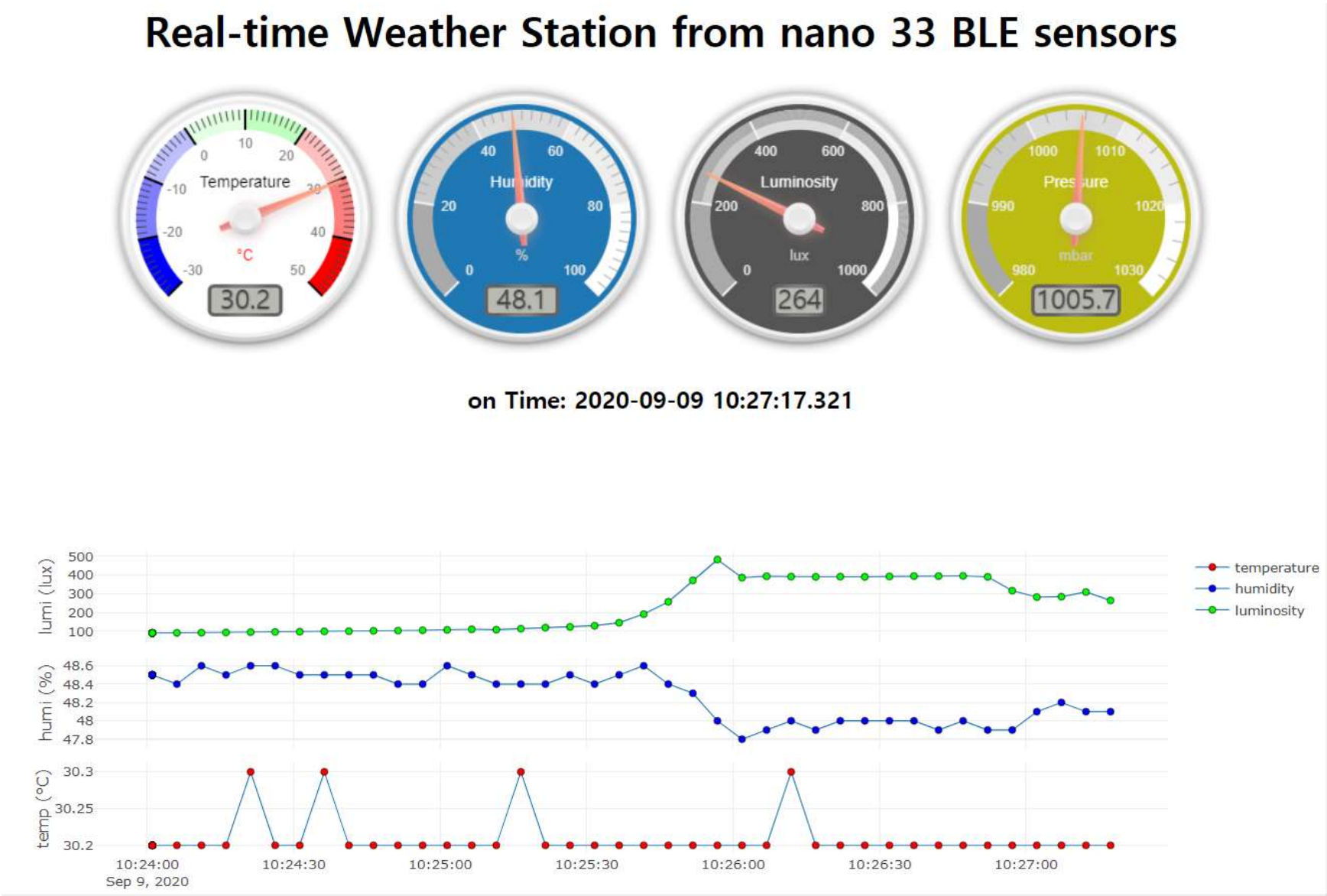
⑦ All *.html in public folder

⑧ Delete 'node_modules' folder

Email : chaos21c@gmail.com

Lecture materials

References & good sites

- ✓ http://www.arduino.cc Arduino Homepage
- ✓ http://www.nodejs.org/ko Node.js
- ✓ https://plot.ly/ plotly
- ✓ https://www.mongodb.com/ MongoDB
- ✓ http://www.w3schools.com By w3schools
- ✓ http://www.github.com GitHub

Target of this class