

A low-angle, upward-looking perspective of several tall, modern skyscrapers with glass facades, creating a sense of height and architectural scale. The buildings are dark and their windows reflect light, creating a grid-like pattern. The sky is a uniform, light gray.


HW5

Download XAMPP

[Apache Friends](#) [下載](#) [Hosting](#) [社區](#) [關於](#)

下載

XAMPP是一個易於安裝的Apache發行版本，其中包含MariaDB、PHP和Perl。僅僅需要下載並啟動安裝程式。就這麼簡單。

 XAMPP for **Windows** 7.4.33, 8.0.25 & 8.1.12

版本	校驗碼	大小
7.4.33 / PHP 7.4.33 包含什麼內容？	md5 sha1	下載 (64 bit) 141 Mb
8.0.25 / PHP 8.0.25 包含什麼內容？	md5 sha1	下載 (64 bit) 143 Mb
8.1.12 / PHP 8.1.12 包含什麼內容？	md5 sha1	下載 (64 bit) 147 Mb

[需求](#) [更多下載](#) »

Windows XP or 2003 are not supported. You can download a compatible version of XAMPP for these platforms [here](#).

Start Apache&MySQL

XAMPP Control Panel v3.3.0 [Compiled: Apr 6th 2021]

XAMPP Control Panel v3.3.0

Modules

Service	Module	PID(s)	Port(s)	Actions
<input type="checkbox"/>	Apache			Start Admin Config Logs
<input type="checkbox"/>	MySQL			Start Admin Config Logs
<input type="checkbox"/>	FileZilla			Start Admin Config Logs
<input type="checkbox"/>	Mercury			Start Admin Config Logs
<input type="checkbox"/>	Tomcat			Start Admin Config Logs

Config Netstat Shell Explorer Services Help Quit

下午 03:55:21 [mysql] Attempting to start MySQL app...
下午 03:55:21 [Apache] Status change detected: running
下午 03:55:21 [mysql] Status change detected: running
下午 02:32:41 [Apache] Attempting to stop Apache (PID: 8952)
下午 02:32:41 [Apache] Attempting to stop Apache (PID: 9040)
下午 02:32:41 [Apache] Status change detected: stopped
下午 02:32:42 [mysql] Attempting to stop MySQL app...
下午 02:32:42 [mysql] Status change detected: stopped

Start Apache&MySQL

XAMPP Control Panel v3.3.0 [Compiled: Apr 6th 2021]

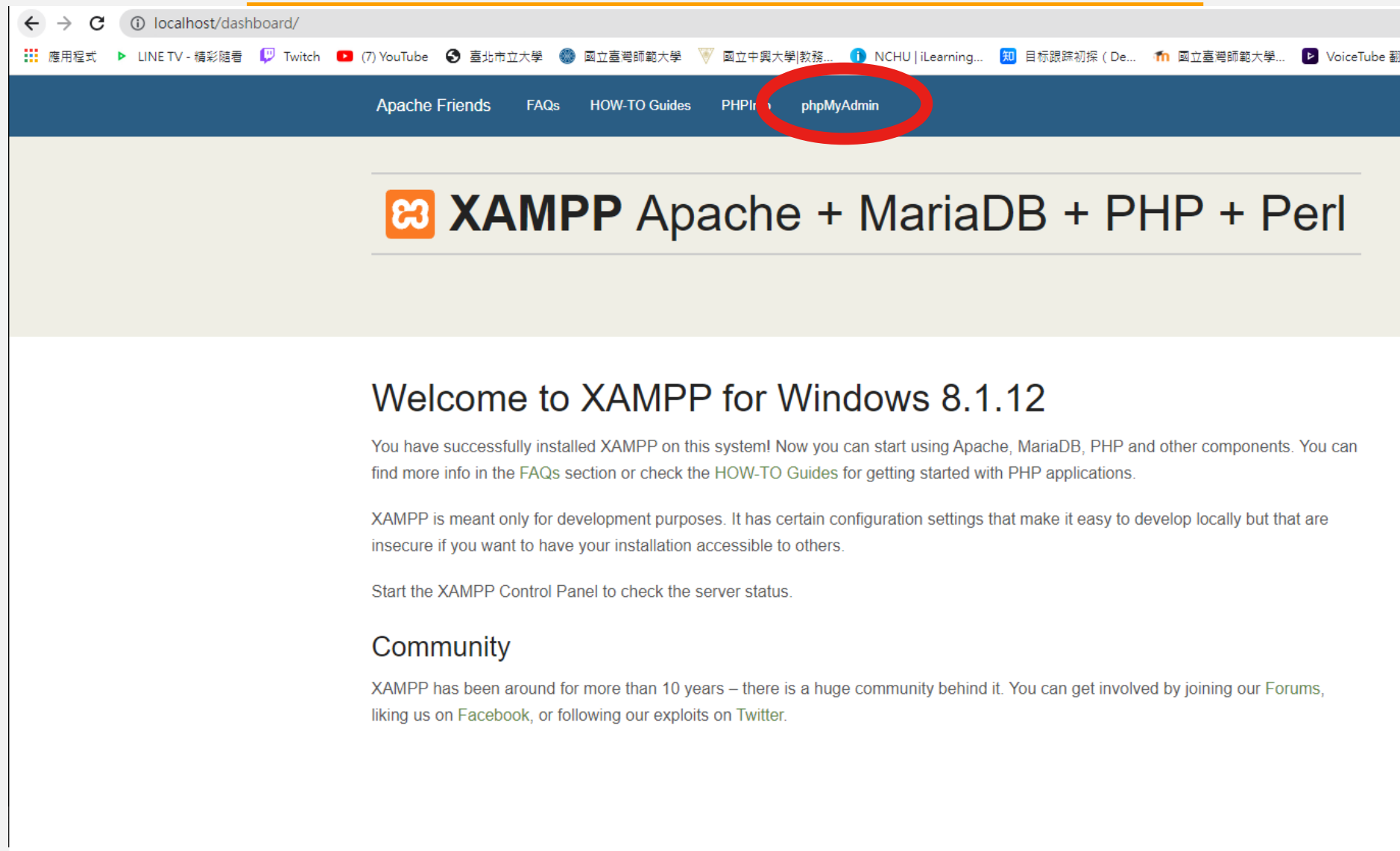
XAMPP Control Panel v3.3.0

Service	Module	PID(s)	Port(s)	Actions
<input type="checkbox"/>	Apache	12696 16268	80, 443	<input type="button" value="Stop"/> <input type="button" value="Admin"/> <input type="button" value="Config"/> <input type="button" value="Logs"/>
<input type="checkbox"/>	MySQL	19232	3306	<input type="button" value="Stop"/> <input type="button" value="Admin"/> <input type="button" value="Config"/> <input type="button" value="Logs"/>
<input type="checkbox"/>	FileZilla			<input type="button" value="Start"/> <input type="button" value="Admin"/> <input type="button" value="Config"/> <input type="button" value="Logs"/>
<input type="checkbox"/>	Mercury			<input type="button" value="Start"/> <input type="button" value="Admin"/> <input type="button" value="Config"/> <input type="button" value="Logs"/>
<input type="checkbox"/>	Tomcat			<input type="button" value="Start"/> <input type="button" value="Admin"/> <input type="button" value="Config"/> <input type="button" value="Logs"/>

Modules

下午 02:32:41 [Apache] Attempting to stop Apache (PID: 9040)
下午 02:32:41 [Apache] Status change detected: stopped
下午 02:32:42 [mysql] Attempting to stop MySQL app...
下午 02:32:42 [mysql] Status change detected: stopped
下午 02:32:47 [Apache] Attempting to start Apache app...
下午 02:32:47 [Apache] Status change detected: running
下午 02:32:48 [mysql] Attempting to start MySQL app...
下午 02:32:48 [mysql] Status change detected: running

Go Localhost&phpMyAdmin



新建使用者帳號

伺服器: 127.0.0.1

資料庫 SQL 狀態 **使用者帳號** 出 匯入 設定 備援 變數 字元集 引!

使用者帳號一覽 使用者群組

使用者帳號一覽

	使用者名稱	主機名稱	密碼	全域權限	使用者群組	允許授權(Grant)	動作
<input type="checkbox"/>	任何	%	否	USAGE		否	編輯權限 匯出 Lock
<input type="checkbox"/>	pma	localhost	否	USAGE		否	編輯權限 匯出 Lock
<input type="checkbox"/>	root	127.0.0.1	否	ALL PRIVILEGES	是		編輯權限 匯出 Lock
<input type="checkbox"/>	root	::1	否	ALL PRIVILEGES	是		編輯權限 匯出 Lock
<input type="checkbox"/>	root	localhost	否	ALL PRIVILEGES	是		編輯權限 匯出 Lock
<input type="checkbox"/>	test123	%	是	ALL PRIVILEGES	是		編輯權限 匯出 Lock
<input type="checkbox"/>	test123	localhost	是	ALL PRIVILEGES	是		編輯權限 匯出 Lock

↑ ☐ 全選 已選擇項目: 匯出

新增

新增使用者帳號

登入資訊

使用者名稱：

使用文字方塊

主機名稱：

任何主機

%

密碼：

使用文字方塊

強度：

重新輸入：

認證外掛程式

原生 MySQL 認證

產生密碼：

產生

使用者帳號的資料庫

☐ 建立與使用者同名的資料庫，並授予所有權限。

☐ 給以 帳號_ 開頭的資料庫 (username_) 授予所有權限。

全域權限

☐ 全選

注意：MySQL 權限名稱會以英文表示。

☐ 資料

☐ SELECT

☐ INSERT

☐ UPDATE

☐ DELETE

☐ FILE

☐ 結構

☐ CREATE

☐ ALTER

☐ INDEX

☐ DROP

☐ CREATE TEMPORARY TABLES

☐ SHOW VIEW

☐ 管理

☐ GRANT

☐ SUPER

☐ PROCESS

☐ RELOAD

☐ SHUTDOWN

☐ SHOW DATABASES

資源限制

注意：設定為 0 即代表無限制。

MAX QUERIES PER HOUR

0

MAX UPDATES PER HOUR

0

MAX CONNECTIONS PER HOUR

0

新建使用者帳號

紅圈填test123

全域權限打勾

登入資訊

使用者名稱：

使用文字方塊

主機名稱：

任何主機

%

密碼：

使用文字方塊

強度：

重新輸入：

認證外掛程式

原生 MySQL 認證

產生密碼：

產生

使用者帳號的資料庫

☐ 建立與使用者同名的資料庫，並授予所有權限。

☐ 給以 帳號_ 開頭的資料庫 (username_) 授予所有權限。

全域權限

☐ 全選

注意：MySQL 權限名稱會以英文表示。

☐ 資料

☐ SELECT

☐ INSERT

☐ UPDATE

☐ DELETE

☐ FILE

☐ 結構

☐ CREATE

☐ ALTER

☐ INDEX

☐ DROP

☐ CREATE TEMPORARY TABLES

☐ SHOW VIEW

☐ 管理

☐ GRANT

☐ SUPER

☐ PROCESS

☐ RELOAD

☐ SHUTDOWN

☐ SHOW DATABASES

資源限制

注意：設定為 0 即代表無限制。

MAX QUERIES PER HOUR

0

MAX UPDATES PER HOUR

0

MAX CONNECTIONS PER HOUR

0

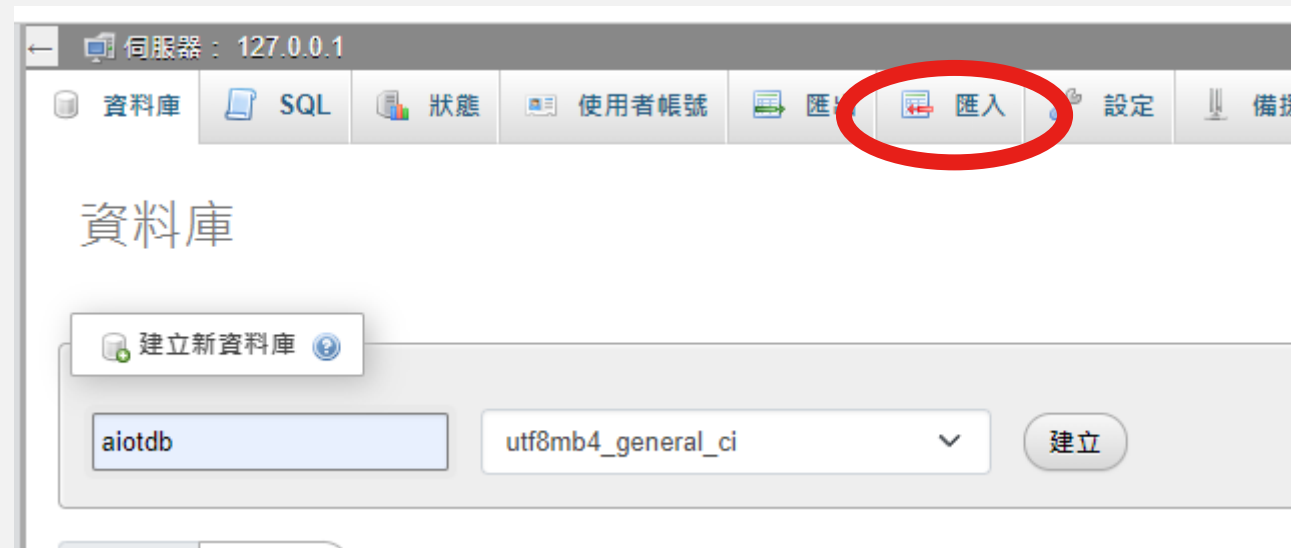
再新建使用者帳號

紅圈填test123

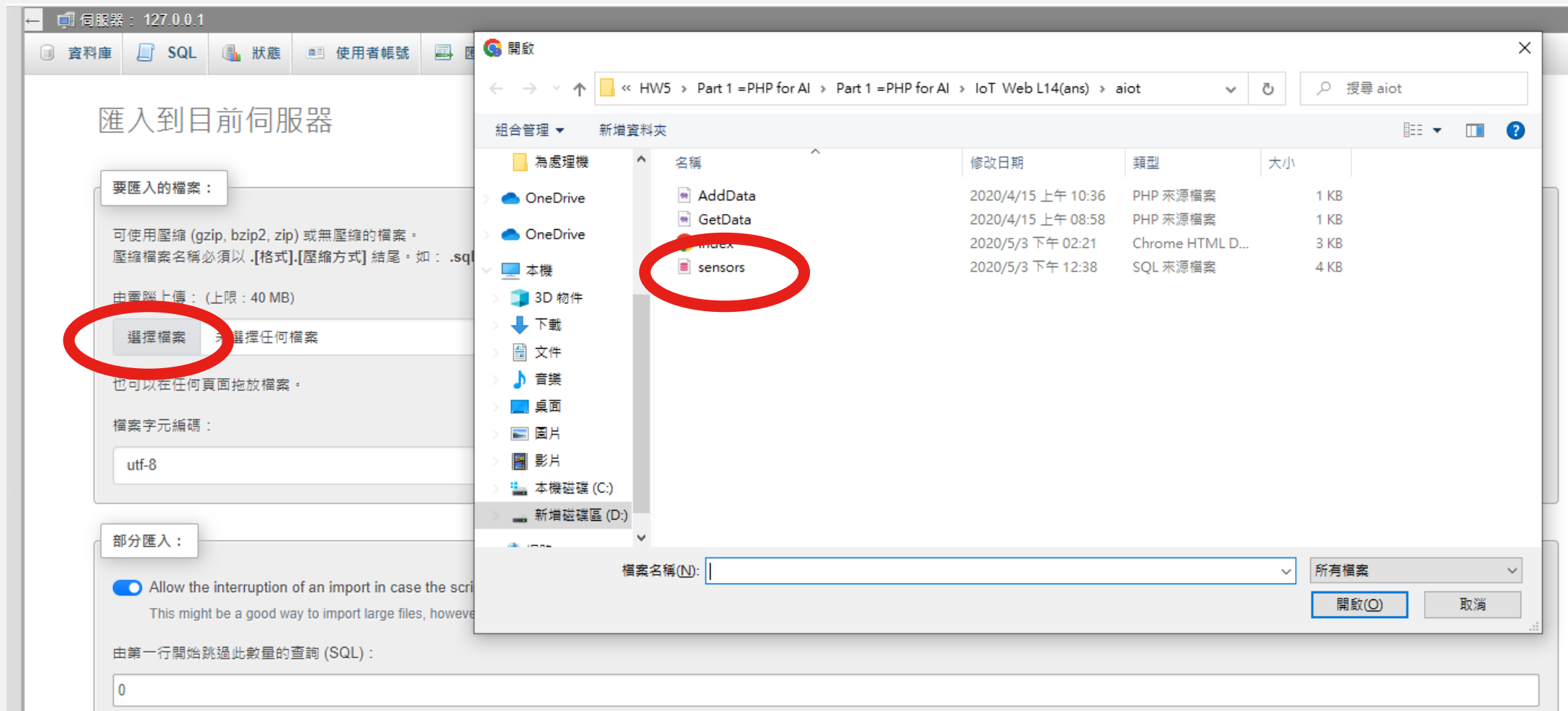
全域權限打勾

主機名稱改主機


到資料庫建立aiotdb資料庫並點匯入



匯入sensors



Download HeidiSQL



HeidiSQL

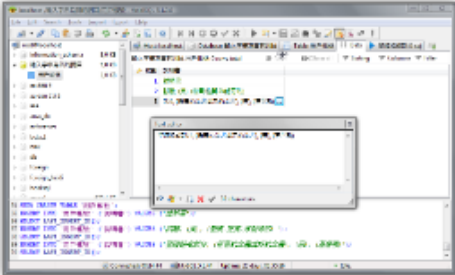
Download the **SQL editor** with the highest user satisfaction* **for free.**

[Home](#) [Downloads](#) [Images](#) [Forum](#) [Donate](#) [Bugtracker](#) [Help](#)

What's this?

HeidiSQL is free software, and has the aim to be easy to learn. "Heidi" lets you see and edit data and structures from computers running one of the database systems MariaDB, MySQL, Microsoft SQL, PostgreSQL and SQLite. Invented in 2002 by Ansgar, HeidiSQL belongs to the most popular tools for MariaDB and MySQL worldwide.

Download HeidiSQL, read further about [features](#), take part in [discussions](#) or see some [screenshots](#).



Opening large files

Selected files have a size of 706.7 MB

Run Hei(s) directly
... without loading into the editor

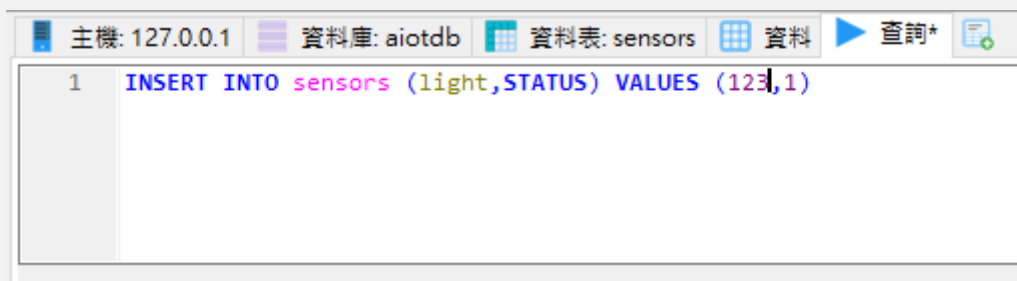
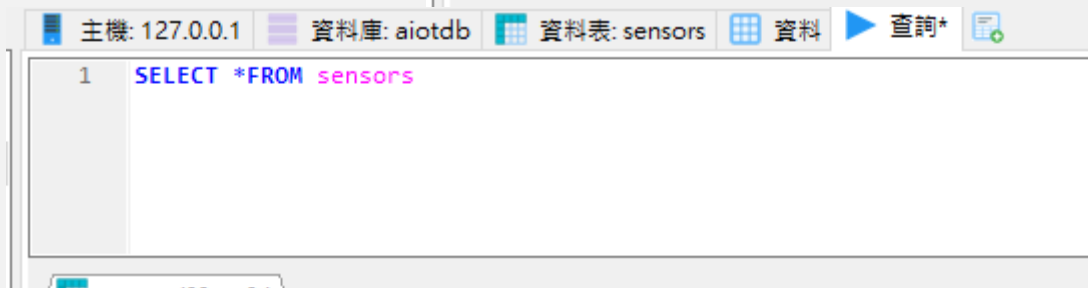
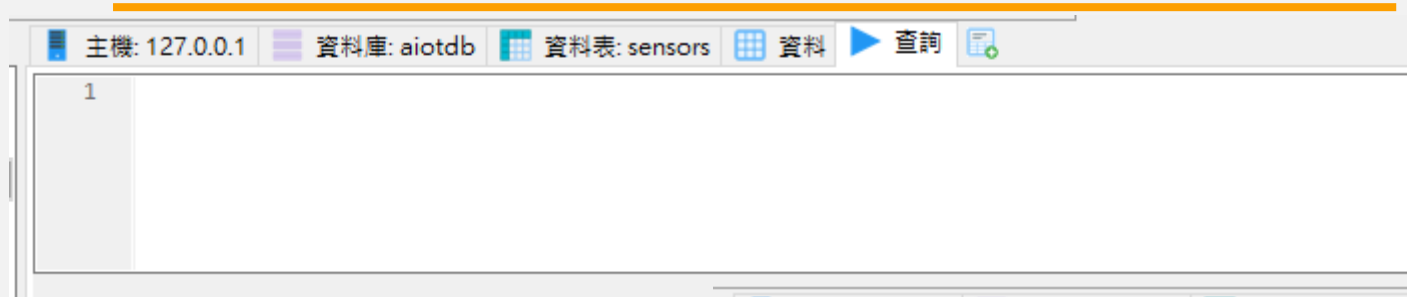
Load Hei(s) into the editor
Can cause large memory usage

Cancel

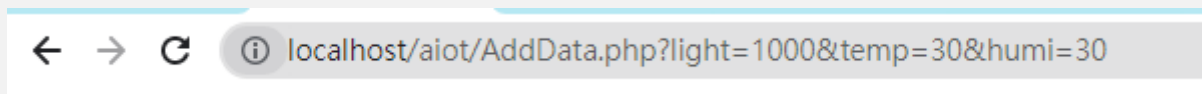
File list

huge-table-inserts.sql (26.3 MB)
many-inserts.sql (15.7 MB)
database-08.sql (272.7 MB)
logging.sql (3.3 MB)
order_report.sql (117.7 MB)

使用查詢



網址上輸入 AddData



55	2022-12-13 13:51:18	1,000	30	30	0
56	2022-12-13 10:52:46	22	0	0	0
57	2022-12-13 10:52:46	33	0	0	0
58	2022-12-13 10:52:46	44	0	0	0
59	2022-12-13 10:52:46	44	44	44	0
60	2022-12-13 13:51:18	1,000	30	30	0
61	2022-12-13 13:51:18	1,000	30	30	0
62	2022-12-13 10:51:00	44	0	0	0
63	2022-12-13 10:52:46	55	0	0	0
64	2022-12-13 13:51:18	555	0	0	1
65	2022-12-13 13:51:18	987	0	0	0
66	2022-12-13 13:51:18	897	87	87	0
67	2022-12-13 14:55:47	500	0	0	1
68	2022-12-13 14:55:58	123	0	0	1
69	2022-12-13 14:59:52	1,000	30	30	0

資料庫尾端

AddData.php

```
1 <?php
2
3
4 //0.get parameters input
5 $light = 0;
6 $humi = 0;
7 $temp = 0;
8
9
10 if (!empty($_GET['light'])) {
11     $light = $_GET['light'];
12 }
13 if (!empty($_GET['humi'])) {
14     $humi = $_GET['humi'];
15 }
16 if (!empty($_GET['temp'])) {
17     $temp = $_GET['temp'];
18 }
19
```

```

20 //1.connctet db
21 $mysqli = new mysqli("localhost","test123","test123","aiotdb");
22
23 // Check connection
24 if ($mysqli -> connect_errno) {
25     echo "Failed to connect to MySQL: " . $mysqli -> connect_error;
26     exit();
27 }
28 else
29 {
30     echo "Succes";
31 }
32
33 //2.query db
34 $sqlquery = "insert into sensors (light,humi,temp) VALUES ($light,$humi,$temp)";
35 $result = $mysqli -> query($sqlquery);
36
```

```
37 //3.close db
38
39 $mysqli -> close();
40
41 header( "Location: http://localhost/aiot/");
42
43 ?>
```

初始化

connect

query

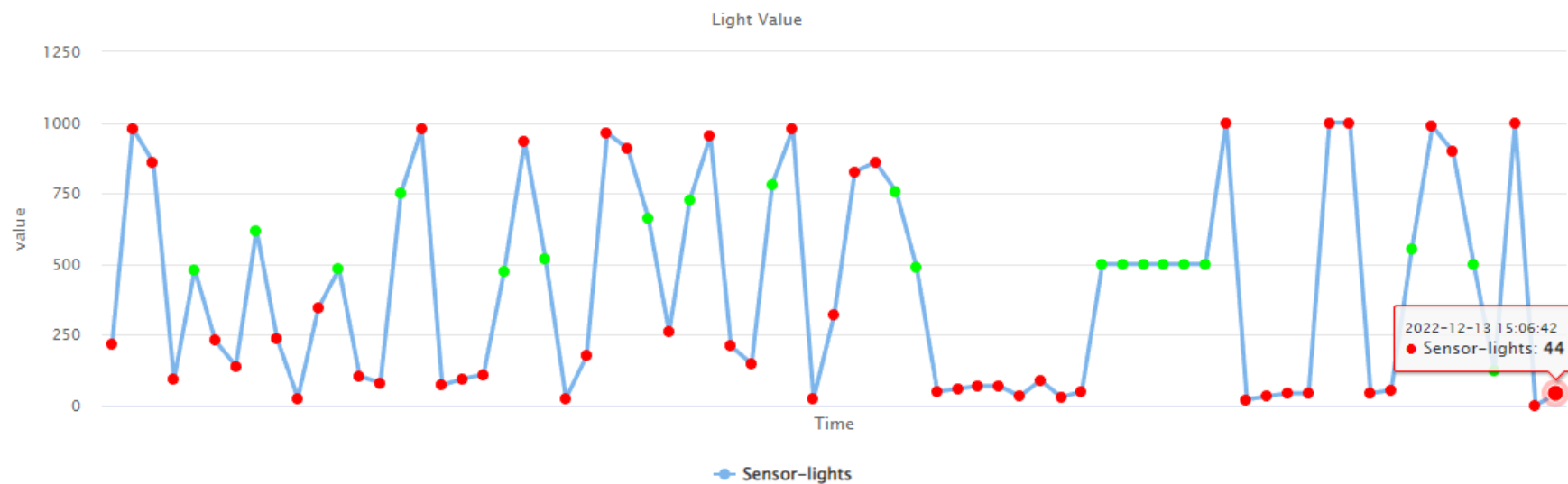
Close 並導向

AddData.php

localhost/aiot/AddData2.php?light=987

應用程式 LINE TV - 精彩隨看 Twitch (7) YouTube 臺北市立大學 國立臺灣師範大學 國立中興大學|教務... NCHU | iLearning... 知 目標跟蹤初探 (De... 國立臺灣師範大學... VoiceTube 翻譯字典 線上遊戲-beanfun!...

Sensor data from MySQL to Highcharts



Highcharts.com

GetData

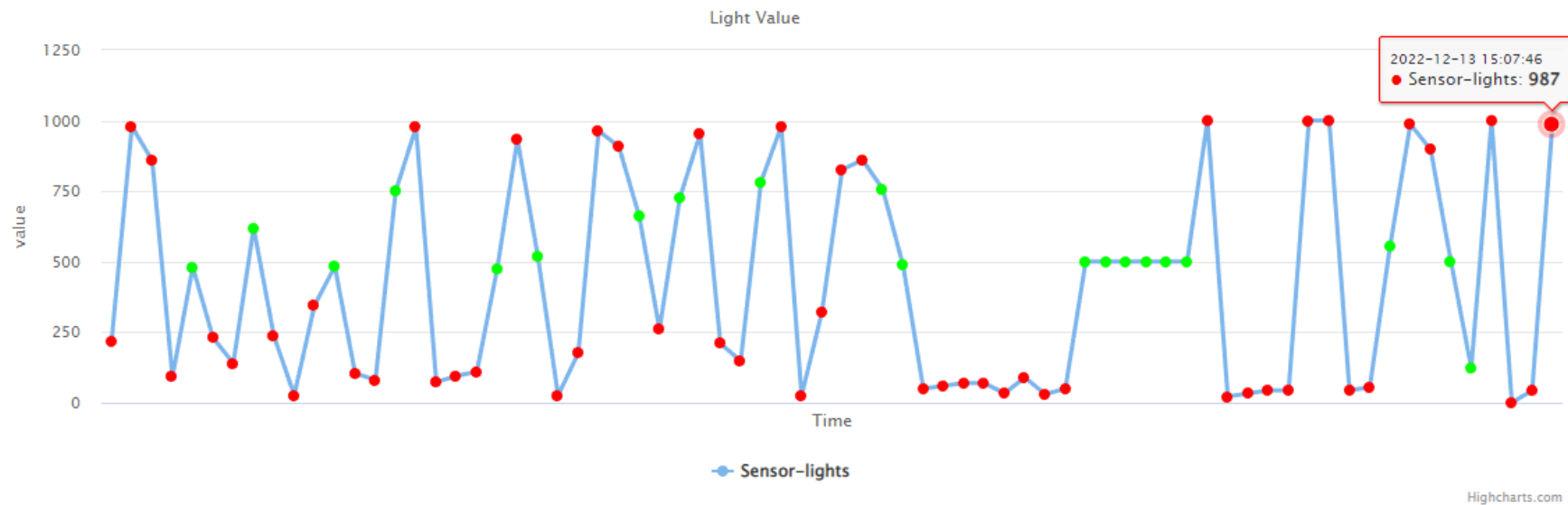
Trigger EA

Trigger NB

Random

AddData.php

Sensor data from MySQL to Highcharts



GetData

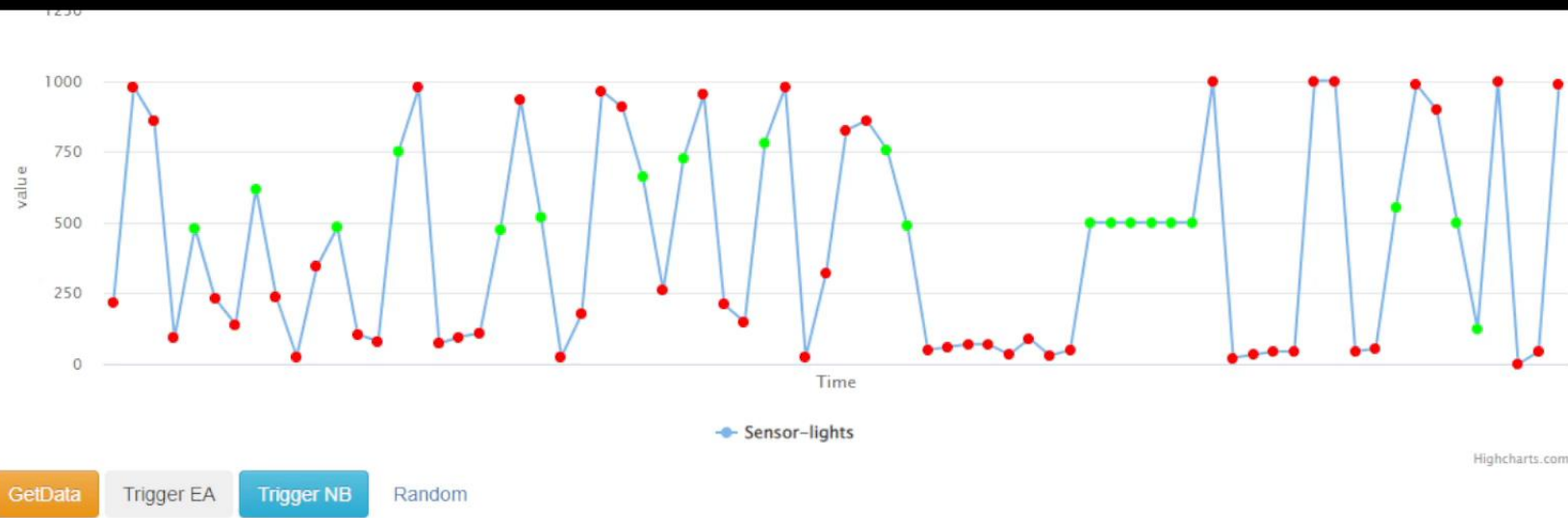
Trigger EA

Trigger NB

Random

Refresh in index.html(Demo highchart)

```
3 <head>  
4   <meta charset="UTF-8" http-equiv="refresh" content="5">  
5  
6 </html>
```



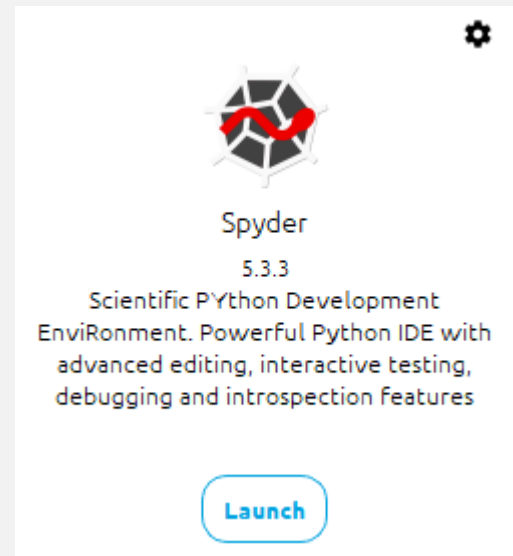
創建環境(Anaconda)

Create new environment

Name:

Location: C:\Users\EE720A\anaconda3\envs\test

Packages: ☒ Python 3.7.15 ☐ R 3.6.1



<input checked="" type="checkbox"/> scikit-learn	A set of python modules for machine learning and data mining	1.0.2
<input checked="" type="checkbox"/> numpy	Array processing for numbers, strings, records, and objects.	1.21.5
<input checked="" type="checkbox"/> pandas	High-performance, easy-to-use data structures and data analysis tools.	1.3.5
<input checked="" type="checkbox"/> pymysql	Pure python mysql driver	1.0.2

Htdos/aiot裡面放AI module

→ ▾ ↑ > 本機 > 本機磁碟 (C:) > xampp > htdocs > aiot >				
★ 快速存取 桌面 下載 文件 圖片 HW4 meeting3 作業考古解答	名稱	修改日期	類型	大小
	.spyproject	2022/12/13 上午 10:40	檔案資料夾	
	AI module (local)	2022/12/13 下午 01:51	檔案資料夾	
	AddData	2020/4/15 上午 10:36	PHP 來源檔案	1 KB
	AddData2	2022/12/13 下午 03:05	PHP 來源檔案	1 KB
	GetData	2020/4/15 上午 08:58	PHP 來源檔案	1 KB
	index	2022/12/13 下午 03:13	Chrome HTML D...	3 KB
	sensors	2020/5/3 下午 12:38	SQL 來源檔案	4 KB

Spyder 編輯 AI module

The screenshot displays the Spyder IDE interface with a dark theme. The main editor window shows a Python script named `myEA.py` located at `C:\xampp\htdocs\aiot\AI module (local)\myEA.py`. The script includes imports for `pandas`, `numpy`, `sklearn`, `mysql.connector`, and `pymysql`. It reads data from `training.csv` and `trainN.csv`, processes it, and connects to a MySQL database.

```
1  -*- coding: utf-8 -*-
2  """
3  Spyder Editor
4
5  This is a temporary script file.
6  """
7  debug = 0
8
9  from pandas import DataFrame as df
10 import pandas as pd # 引用套件並縮寫為 pd
11 import numpy as np
12 #from sklearn import linear_model as lm
13
14 from sklearn.linear_model import LogisticRegression as LR
15 from sklearn import svm
16
17 #import mysql.connector
18 #from sqlalchemy.types import NVARCHAR, Float, Integer
19 #from sqlalchemy import create_engine
20 #import sqlalchemy
21
22 myserver = "localhost"
23 myuser = "test123"
24 mypassword = "test123"
25 mydb = "aiotdb"
26
27
28 #=====read Data=====#
29 #data=pd.read_csv("training.csv") #載入training.csv至data
30 data=pd.read_csv("trainN.csv") #非線性
31
32 X=data['value'].values.reshape(-1,1)
33 Y=data['status'].values.reshape(-1,1)
34
35 #print(X.shape,type(X))
36
37 #將Y轉成Numpy的Array (tuple型態)
38 #=====read Data End=====#
39
40 #model=LR() #邏輯迴歸
41 model = svm.SVC() #非線性
42 #model.fit(ndX,ndY) #訓練模型
43 model.fit(X,Y)
44 my_score=model.score(X,Y)
45 #print(my_score)
46
47
48
49
50 import pymysql.cursors
51 #db = mysql.connector.connect(host="140.120.15.45",user="toto321", passwd="12345678", db="lightdb")
52 #conn = mysql.connector.connect(host=myserver,user=myuser, passwd=mypassword, db=mydb)
53 conn = pymysql.connect(host=myserver,user=myuser, passwd=mypassword, db=mydb)
54
55 c = conn.cursor()
56 if debug:
57     input("pause conn.cursor() ok")
```

The right sidebar contains a **Usage** panel with the following text:

Here you can get help of any object by pressing **Ctrl+I** in front of it, either on the Editor or the Console.

Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in **Preferences > Help**.

[New to Spyder? Read our tutorial](#)

Below the Usage panel is the **Console** panel, which shows the output of the script. It displays a table of data and a warning message.

id	time	value	temp	humi	status
1	2022-12-23 13:54:22	219	26.0	99.0	0
2	2022-12-23 13:52:55	979	70.0	58.0	0
3	2022-12-23 13:52:55	859	12.0	89.0	0
4	2022-12-23 12:12:22	92	93.0	42.0	0
5	2022-12-23 13:52:55	481	42.0	35.0	1
6	2022-12-23 12:12:22	230	5.0	31.0	0
7	2022-12-23 12:10:12	140	83.0	70.0	0
8	2022-12-23 13:52:55	620	47.0	94.0	1
9	2022-12-23 13:54:22	236	19.0	23.0	0
10	2022-12-23 12:12:22	27	78.0	60.0	0

The console also shows a warning message:

```
[5, 8, 12, 15, 20, 22, 27, 29, 33, 39, 40, 49, 50, 51, 52, 53, 54, 64, 67]
C:\Users\EE720A\anaconda3\envs\QQ\lib\site-packages\sklearn\utils\validation.py:993: DataConversionWarning: A
column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example
using ravel().
y = column_or_1d(y, warn=True)
```

The bottom status bar indicates the current file is `myEA.py` and the console is active.

AI module訓練(myEA.py)

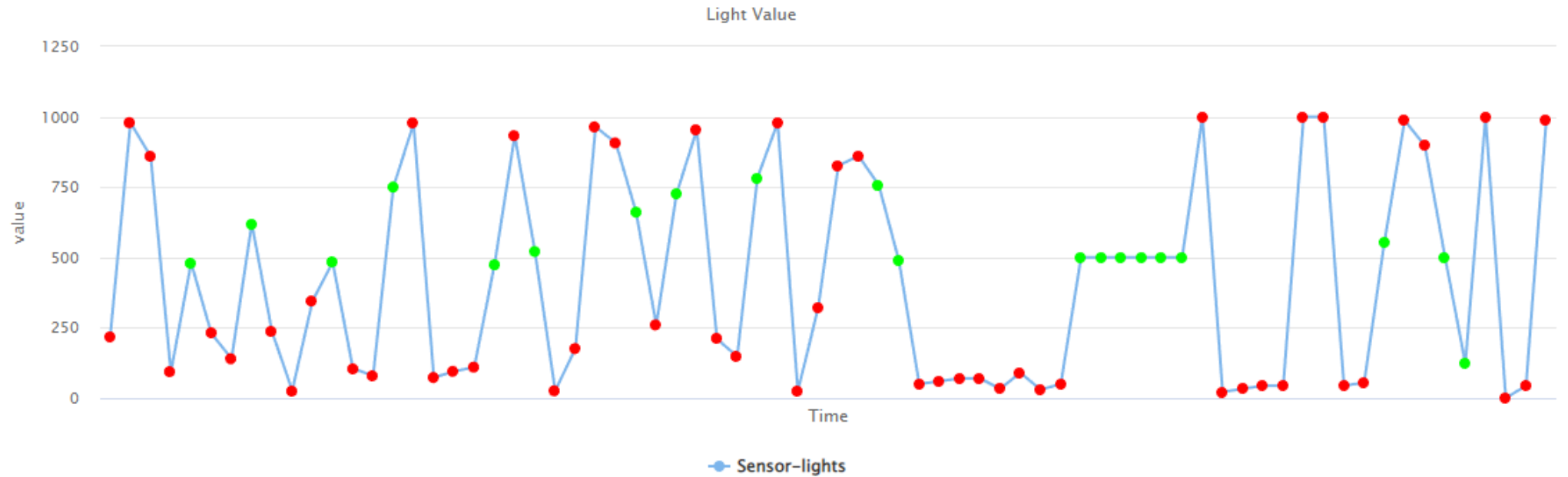
```
from sklearn import svm
```

```
#data=pd.read_csv( 'training.csv' ) #輸入training  
data=pd.read_csv("trainN.csv") #非線性  
  
X=data[ 'value' ].values.reshape(-1,1)  
Y=data[ 'status' ].values.reshape(-1,1)  
  
#print(X.shape,type(X))
```

```
#model=LR() #邏輯迴歸  
model = svm.SVC(k) #非線性  
#model.fit(ndX,ndY) #訓練模型  
model.fit(X,Y)  
my_score=model.score(X,Y)  
#print(my_score)
```

Demo(AI Module & Web)

Sensor data from MySQL to Highcharts



Highcharts.com

GetData

Trigger EA

Trigger NB

Random

pickle

1. 使用 pickle 儲存模型

```
import pickle
with open('./model/xgboost-iris.pickle', 'wb') as f:
    pickle.dump(xgboostModel, f)
```

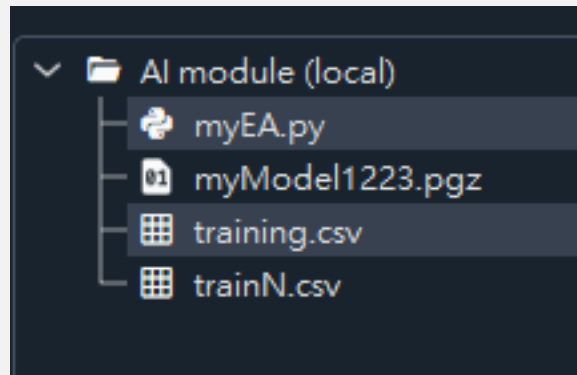
2. 使用 pickle 儲存模型並利用 gzip 壓縮

```
import pickle
import gzip
with gzip.GzipFile('./model/xgboost-iris.pgz', 'w') as f:
    pickle.dump(xgboostModel, f)
```

pickle (myEA.py)

```
print("my model is",model)
import pickle
import gzip
with gzip.GzipFile('myModel1223.pgza', 'w') as f:
    pickle.dump(model, f)
```

打包出去



在fit之後

pickle (myEA.py)

```
import pickle
import gzip
with gzip.open('./myModel1223.pgz', 'r') as f:
    model = pickle.load(f)
```

Load進來

```
#=====read Data=====#
#data=pd.read_csv("training.csv")      #載入training.csv至data
#data=pd.read_csv("trainN.csv")        #非線性

#X=data['value'].values.reshape(-1,1)
#Y=data['status'].values.reshape(-1,1)

#print(X.shape,type(X))

#將sY轉成Numpy的Array (tuple型態)
#=====read Data End=====#

#model=LR()      #邏輯迴歸
#model = svm.SVC() #非線性
#model.fit(ndX,ndY)      #訓練模型
#model.fit(X,Y)

print("my model is",model)
#import pickle
#import gzip
#with gzip.GzipFile('myModel1223.pgz', 'w') as f:
#    pickle.dump(model, f)

#my_score=model.score(X,Y)
#print(my_score)
```

不需要model跟train data

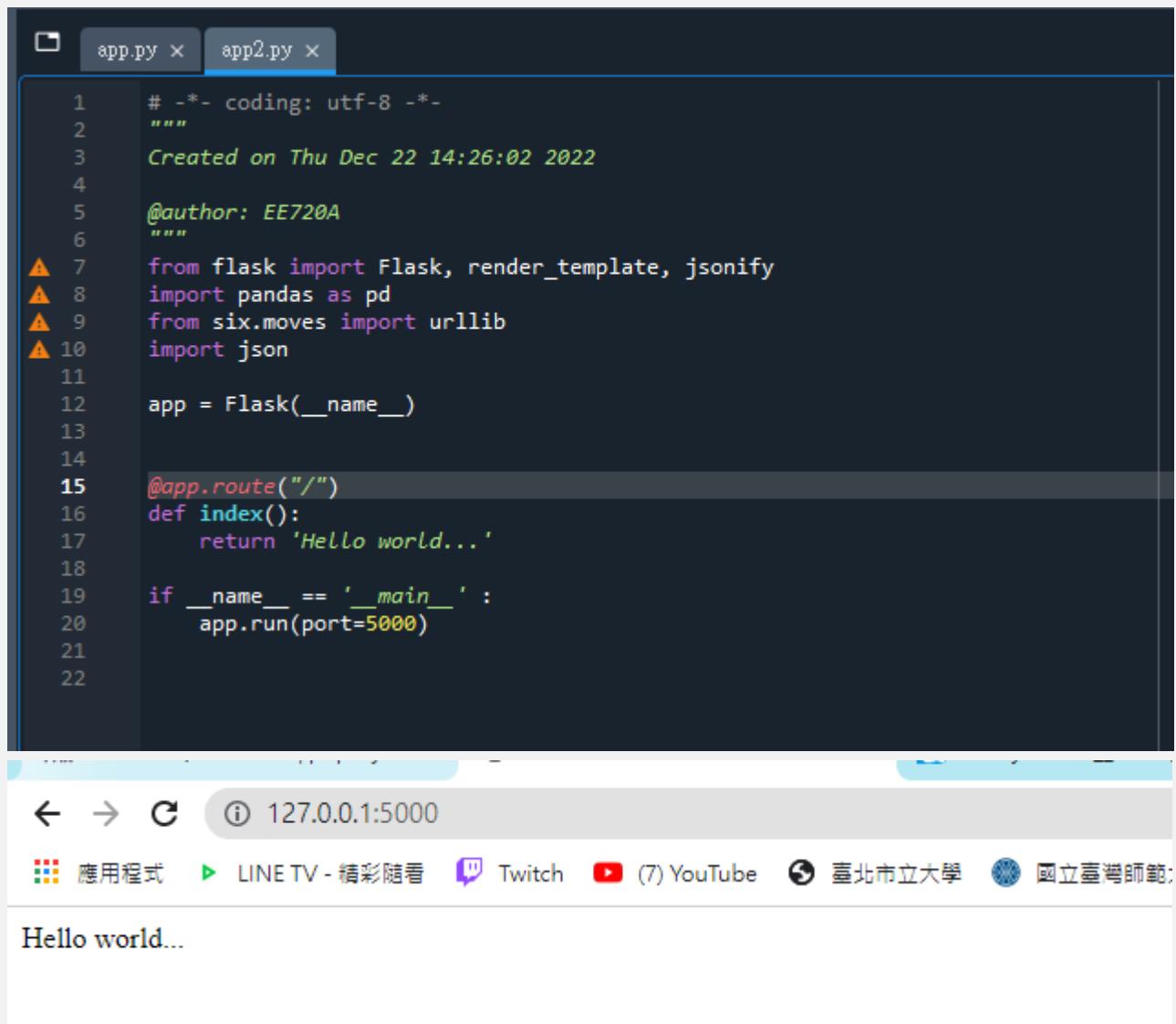
補充commit(有動到資料庫要使用)

```
## make all status =0  
c.execute('update sensors set status=0 where Light>0')  
conn.commit()
```

flask

名稱				
名稱	修改日期	類型	大小	
.spyproject	2022/12/22 下午 01:54	檔案資料夾		
model	2022/12/22 下午 01:49	檔案資料夾		
templates	2022/12/22 下午 01:49	檔案資料夾		
app	2021/5/22 上午 08:22	Python File	3 KB	
app2	2022/12/22 下午 02:32	Python File	1 KB	
deploy_heroku	2021/5/22 上午 08:22	Windows 批次檔案	1 KB	
Procfile	2021/5/22 上午 08:22	檔案	1 KB	
requirements	2021/5/22 上午 08:22	文字文件	1 KB	
runtime	2021/5/22 上午 08:22	文字文件	1 KB	
update_heroku	2021/5/22 上午 08:22	Windows 批次檔案	1 KB	

App2練習



The image shows a code editor window with two tabs: 'app.py' and 'app2.py'. The 'app2.py' tab is active, displaying a Python script for a Flask web application. The script includes a docstring with metadata, imports for Flask, pandas, urllib, and json, and a simple index route that returns 'Hello world...'. The script is run on port 5000. Below the code editor, a web browser window is visible, showing the address '127.0.0.1:5000' and the output 'Hello world...'. The browser's taskbar at the bottom shows various application icons.

```
1  # -*- coding: utf-8 -*-
2  """
3  Created on Thu Dec 22 14:26:02 2022
4
5  @author: EE720A
6  """
7  from flask import Flask, render_template, jsonify
8  import pandas as pd
9  from six.moves import urllib
10 import json
11
12 app = Flask(__name__)
13
14
15 @app.route("/")
16 def index():
17     return 'Hello world...'
18
19 if __name__ == '__main__':
20     app.run(port=5000)
21
22
```

127.0.0.1:5000

Hello world...

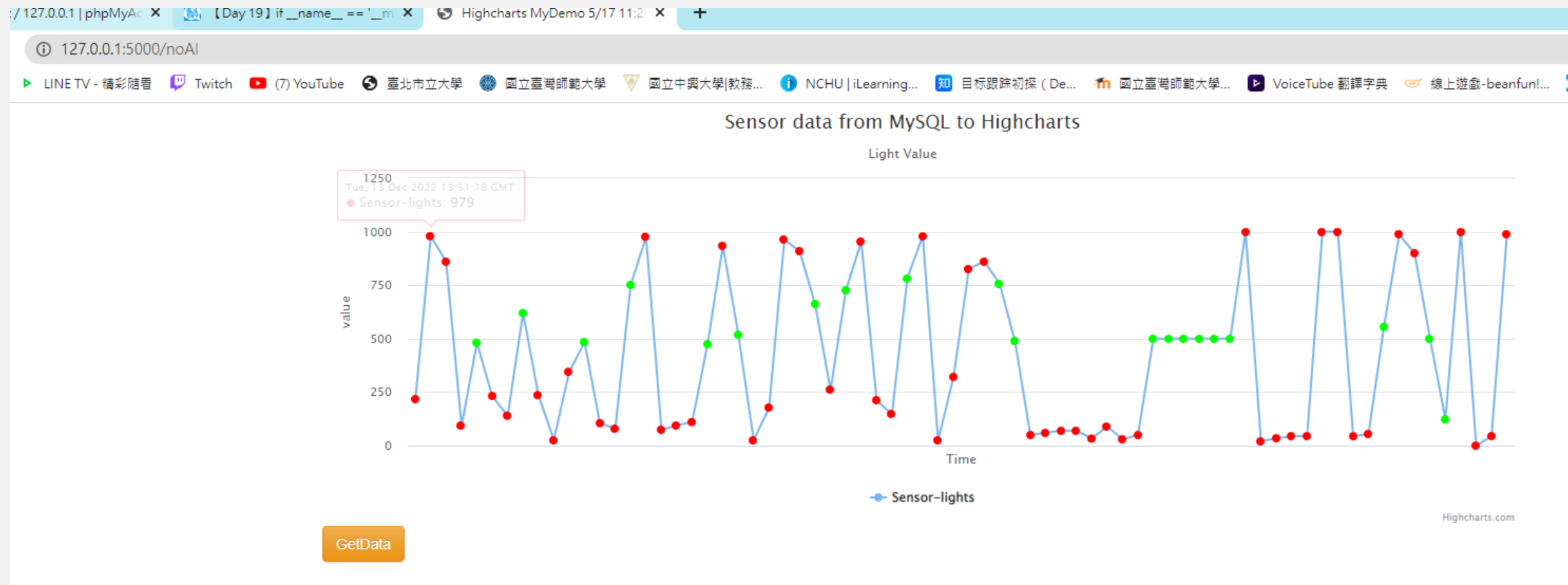
noAI練習

```
...Basics\Part 2=From PHP to Flask Basics\step4_flask_AIoT-20211103T030638Z-001\step4_flask_AIoT\app.py

app.py x indexNoAI.html x

1  from flask import Flask, render_template, jsonify
2  import pandas as pd
3  from six.moves import urllib
4  import json
5
6  app = Flask(__name__)
7
8  @app.route("/data.json")
9  def data():
10     timeInterval = 1000
11     data = pd.DataFrame()
12     featureList = ['market-price',
13                  'trade-volume']
14     for feature in featureList:
15         url = "https://api.blockchain.info/charts/" + feature + "?timespan="+str(timeInterval)+"days&format=j
16         data['time'] = pd.DataFrame(json.loads(urllib.request.urlopen(url).read().decode('utf-8'))['value
17         data[feature] = pd.DataFrame(json.loads(urllib.request.urlopen(url).read().decode('utf-8'))['valu
18     result = data.to_dict(orient='records')
19     seq = [[item['time'], item['market-price'], item['trade-volume']] for item in result]
20     return jsonify(seq)
21
22 @app.route("/")
23 def index():
24     return render_template('index.html')
25
26 @app.route("/noAI")
27 def noAI():
28     return render_template('indexNoAI.html')
29
```

noAI練習



getPredict

value改light

```
app.py x indexNoAI.html x indexAI.html x EA.py x
77 @app.route("/getPredict")
78 def getPredict():
79     debug = 0
80     from pandas import DataFrame as df
81     import pandas as pd          # 引用套件並縮寫為 pd
82     import numpy as np
83     #from sklearn import linear_model as lm
84     from sklearn.linear_model import LogisticRegression as LR
85     #import mysql.connector
86     #from sqlalchemy.types import NVARCHAR, Float, Integer
87     #from sqlalchemy import create_engine
88     #import sqlalchemy
89
90     myserver = "localhost"
91     myuser = "test123"
92     mypassword = "test123"
93     mydb = "aiotdb"
94
95     #===== load model =====
96     import pickle
97     import gzip
98
99     #讀取Model
100     with gzip.open('./model/myModel.pgz', 'r') as f:
101         model = pickle.load(f)
102
103     #print(my_score)
104
105
106
107     import pymysql.cursors
108     #db = mysql.connector.connect(host="140.120.15.45",user="toto321", passwd="12345678", db="ligh
109     #conn = mysql.connector.connect(host=myserver,user=myuser, passwd=mypassword, db=mydb)
110     conn = pymysql.connect(host=myserver,user=myuser, passwd=mypassword, db=mydb)
```

```
print(test_df.head(10))

if debug:
    input("pause.. now show correct one above.....")

#####
...
##Example 1 ## write back mysql #####
threshold = 100
c.execute('update light set status=0 where value>'+str(threshold))
conn.commit()
#results = c.fetchall()
#print(type(results))
#print(results[:10])
input("pause ....update ok.....")
...

##Example 2 ## write back mysql #####
## make all status =0
c.execute('update sensors set status=0 where light=0')

## choose status ==1 have their id available
id_list = list(test_df[test_df['status']==1].id)
print(id_list)

for _id in id_list:
    #print('update light set status=1 where id=='+str(_id))
    c.execute('update sensors set status=1 where id='+str(_id))

conn.commit()

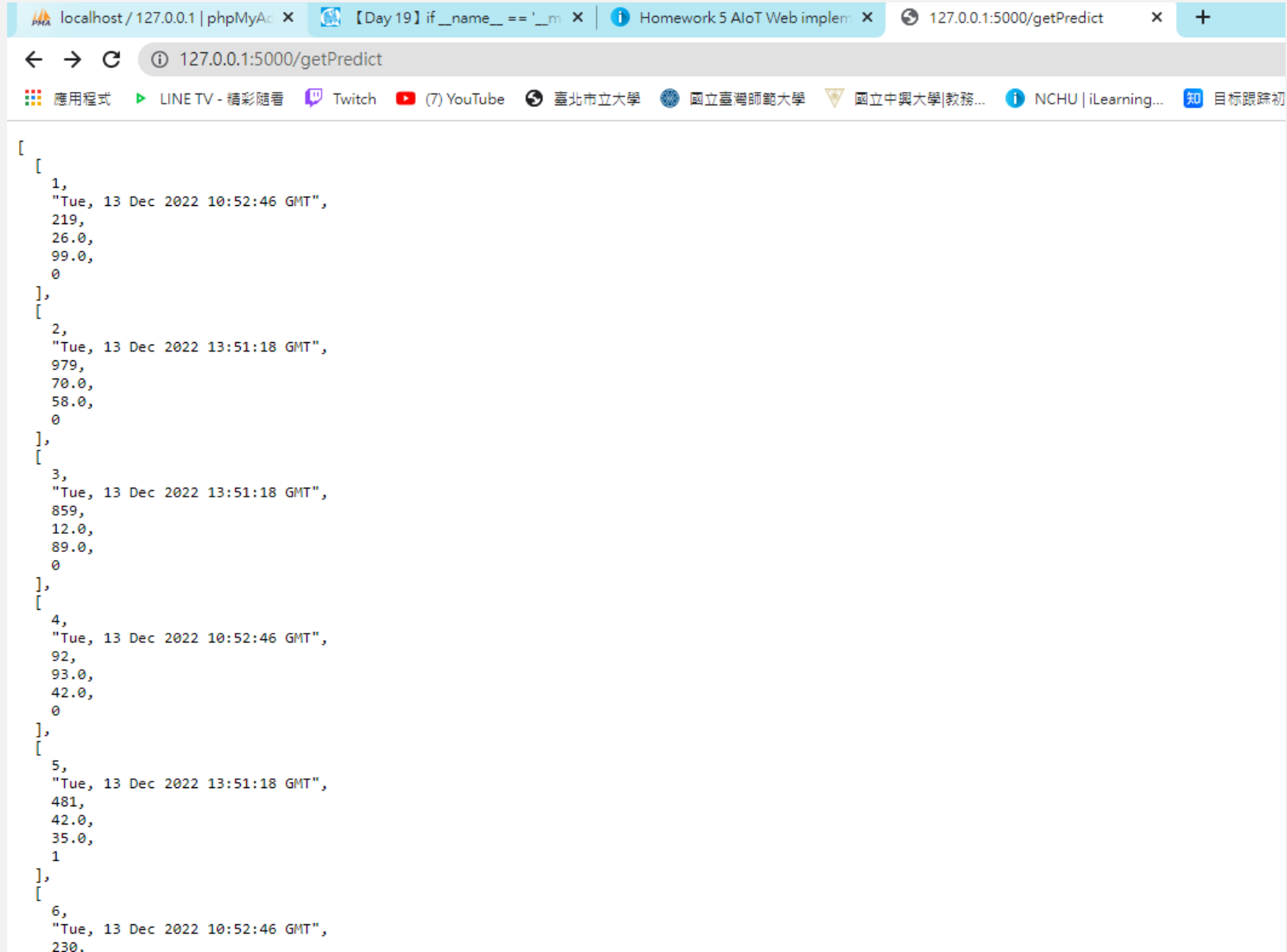
if debug:
    input("pause ....update ok.....")

##### cursor close, conn close
c.close()
conn.close()

test_df = df(list(results), columns=['id', 'time', 'value', 'temp', 'humi', 'status'])

print(test_df.head(10))
result = test_df.to_dict(orient='records')
seq = [[item['id'], item['time'], item['value'], item['temp'], item['humi'], item['status']] for item in result]
return jsonify(seq)
```

getPredict

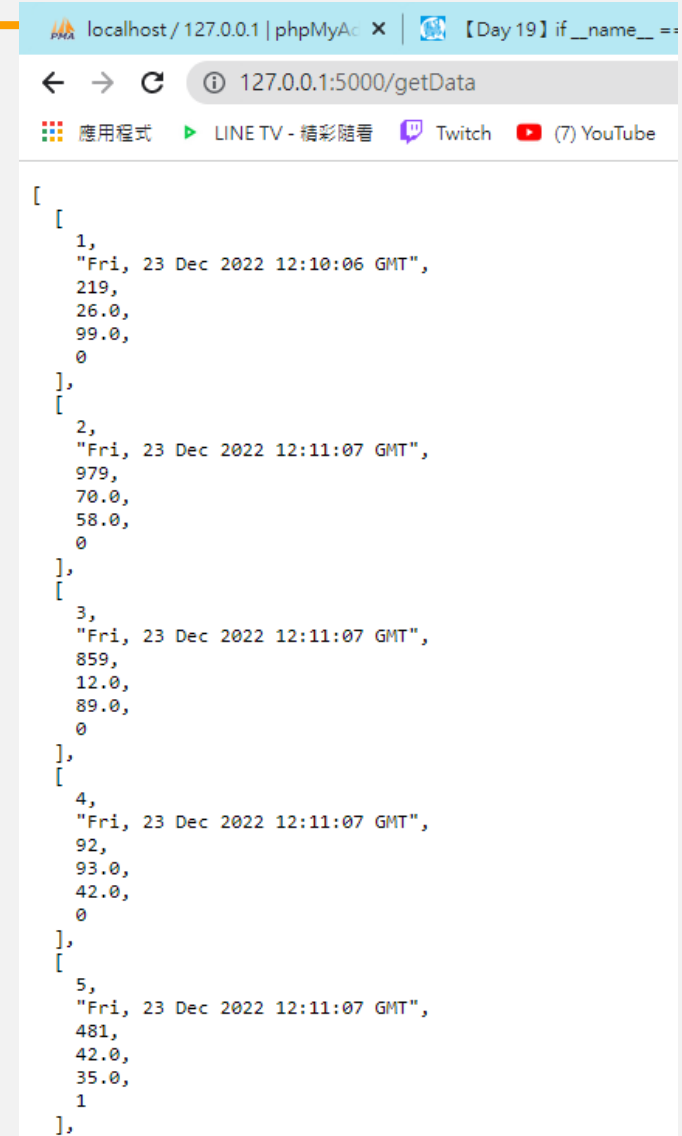


```
[
  [
    1,
    "Tue, 13 Dec 2022 10:52:46 GMT",
    219,
    26.0,
    99.0,
    0
  ],
  [
    2,
    "Tue, 13 Dec 2022 13:51:18 GMT",
    979,
    70.0,
    58.0,
    0
  ],
  [
    3,
    "Tue, 13 Dec 2022 13:51:18 GMT",
    859,
    12.0,
    89.0,
    0
  ],
  [
    4,
    "Tue, 13 Dec 2022 10:52:46 GMT",
    92,
    93.0,
    42.0,
    0
  ],
  [
    5,
    "Tue, 13 Dec 2022 13:51:18 GMT",
    481,
    42.0,
    35.0,
    1
  ],
  [
    6,
    "Tue, 13 Dec 2022 10:52:46 GMT",
    230,
```


getRandom

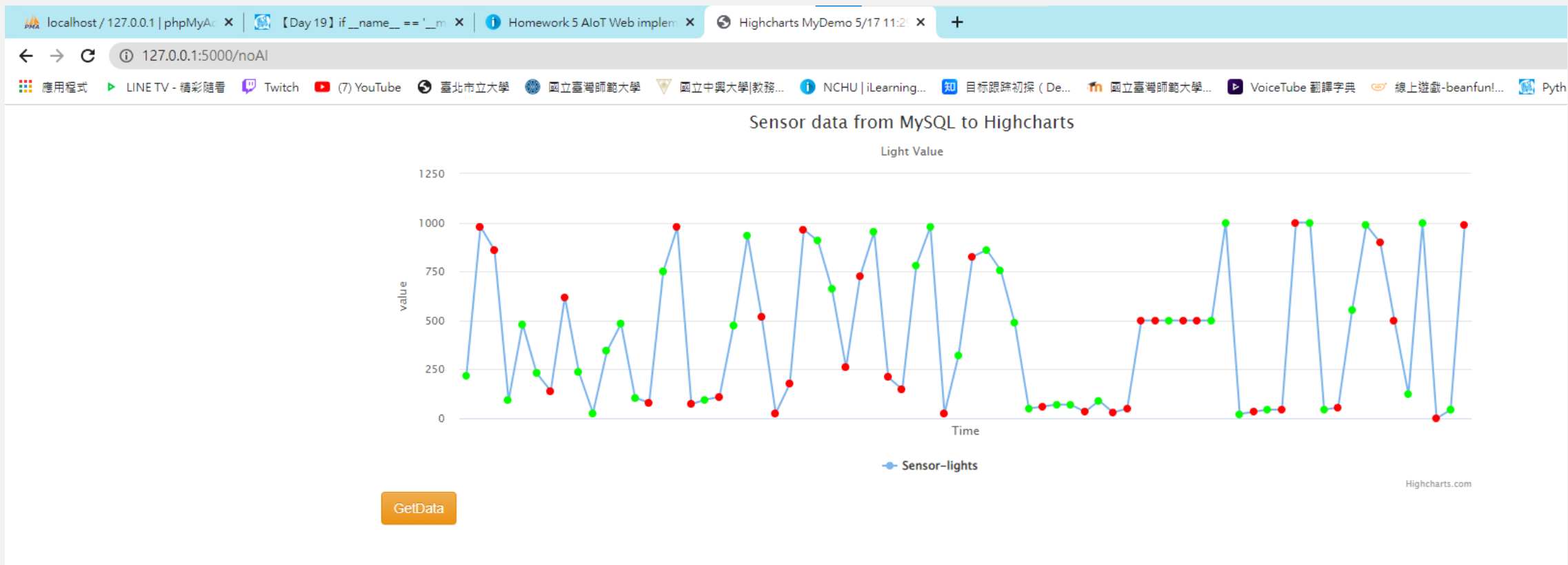
```
@app.route("/getData")
def setRandom():
    myserver = "localhost"
    myuser = "test123"
    mypassword = "test123"
    mydb = "aiotdb"
```

```
c.execute("update sensors set status = RAND() where true")
conn.commit()
c.execute("select * from sensors")
```



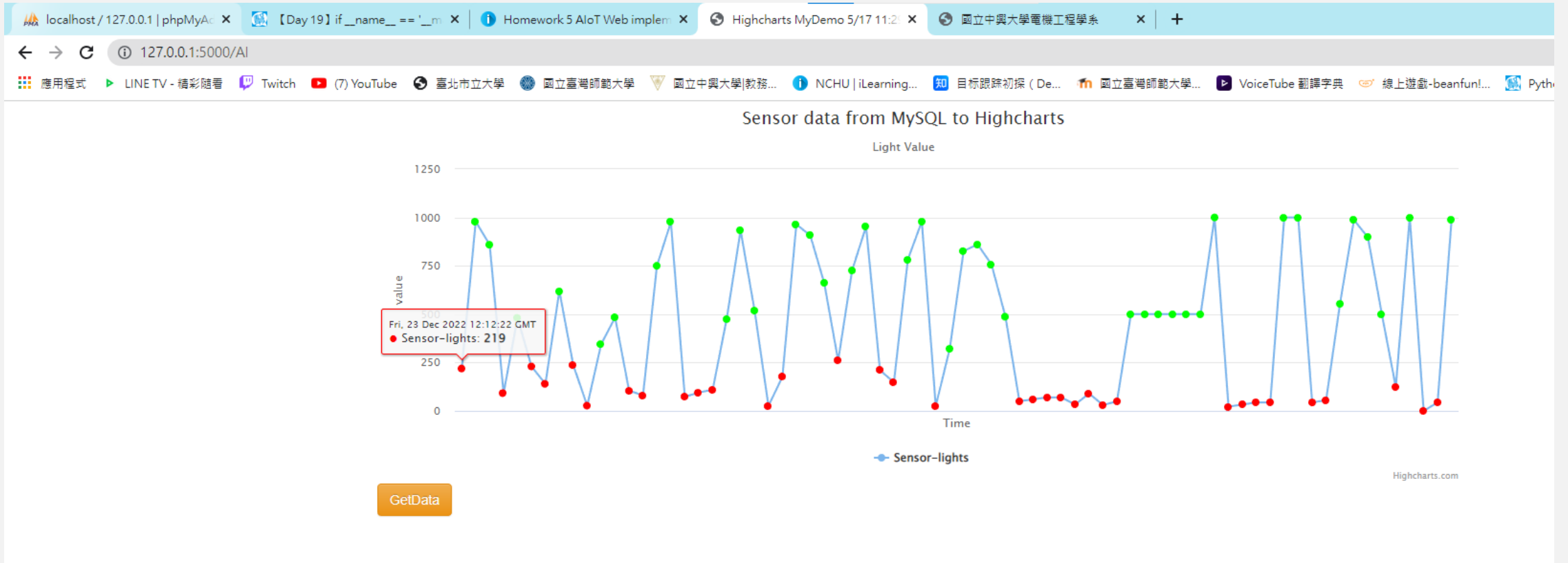
```
[
  [
    1,
    "Fri, 23 Dec 2022 12:10:06 GMT",
    219,
    26.0,
    99.0,
    0
  ],
  [
    2,
    "Fri, 23 Dec 2022 12:11:07 GMT",
    979,
    70.0,
    58.0,
    0
  ],
  [
    3,
    "Fri, 23 Dec 2022 12:11:07 GMT",
    859,
    12.0,
    89.0,
    0
  ],
  [
    4,
    "Fri, 23 Dec 2022 12:11:07 GMT",
    92,
    93.0,
    42.0,
    0
  ],
  [
    5,
    "Fri, 23 Dec 2022 12:11:07 GMT",
    481,
    42.0,
    35.0,
    1
  ],
]
```

noAI(Demo flask)



有隨機的noAI

AI (Demo flask)



```
##example 2 ## write back mysql #####  
## make all status =0  
c.execute('update sensors set status=0 where true')  
## choose status =1 have their id available
```

有隨機的AI

Ngrok(install)

(37) 2022 12 21 09 2 x | 快速讓外網連接本機 x | ngrok - download x | ngrok教學-從外部連 x | ERR_NGROK_6022 x | Setup - ngrok x | ERR_NGROK_6022 x | Chart x | +

ngrok.com/download

Announcing 50M round for API-first ingress-as-a-service. [Learn more](#)

ngrok

Product Solutions Customers Docs Pricing Download Login Sign up

Install ngrok

The fastest way to put anything on the internet.

Mac OS Windows Linux Docker Other

Install ngrok via Chocolatey

```
$ choco install ngrok
```

or


Download ZIP file

Windows (64-bit) Download

Double-click .zip file once downloaded

Ask a question

Connect account

 ngrok	2022/9/15 上午 01:26	應用程式	18,917 KB
---	--------------------	------	-----------

2. Connect your account

Running this command will add your authtoken to the default `ngrok.yml` configuration file. This will grant you access to more features and longer session times. Running tunnels will be listed on the [endpoints page](#) of the dashboard.

```
$ ngrok config add-authtoken 2JJ1C4ngBebBXKptL31Ep3QW0ZV_2GjyFCXcYZWZp4RqwTcCn
```

```
D:\AIOT\HW5\ngrok-v3-stable-windows-amd64>ngrok config add-authtoken 2JJ1C4ngBebBXKptL31Ep3QW0ZV_2GjyFCXcYZWZp4RqwTcCn
Authtoken saved to configuration file: C:\Users\EE720A\AppData\Local\ngrok\ngrok.yml
```

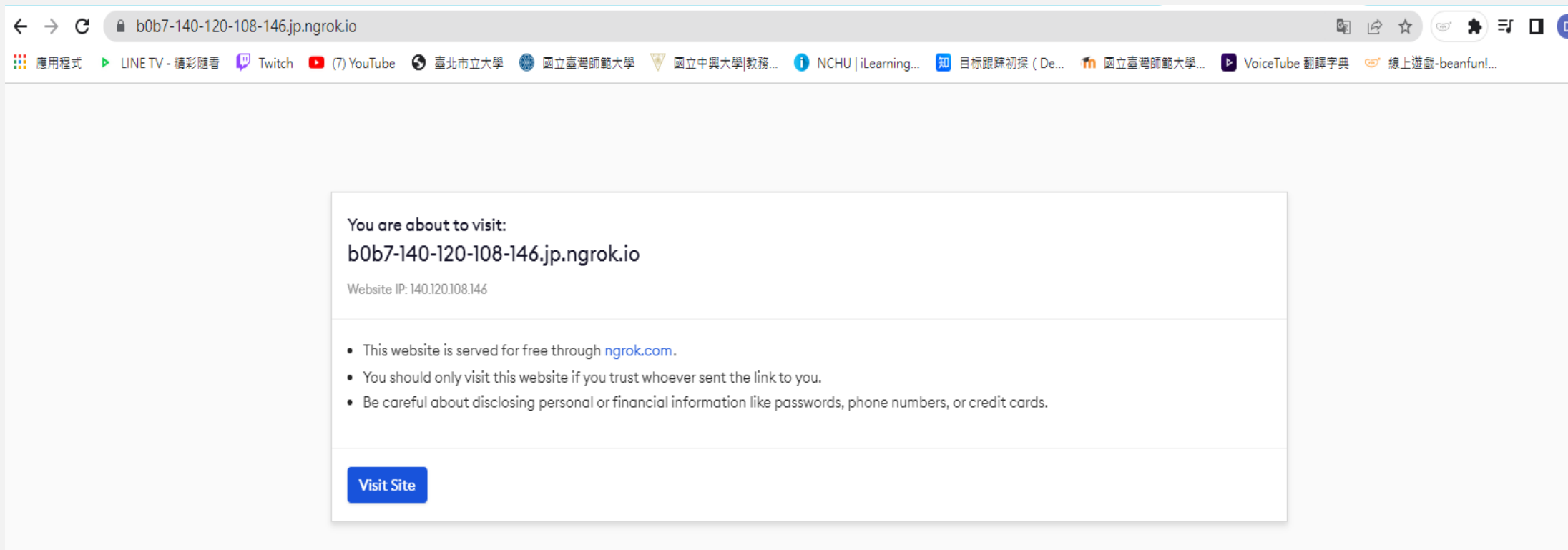
Connect port

```
D:\AIOT\HW5\ngrok-v3-stable-windows-amd64>ngrok http 5000
```

```
ngrok (Ctrl+C to quit)
Try our new native Go library: https://github.com/ngrok/ngrok-go
Session Status      online
Account             love759542@gmail.com (Plan: Free)
Version             3.1.0
Region              Japan (jp)
Latency              -
Web Interface        http://127.0.0.1:4041
Forwarding           https://b0b7-140-120-108-146.jp.ngrok.io -> http://localhost:5000

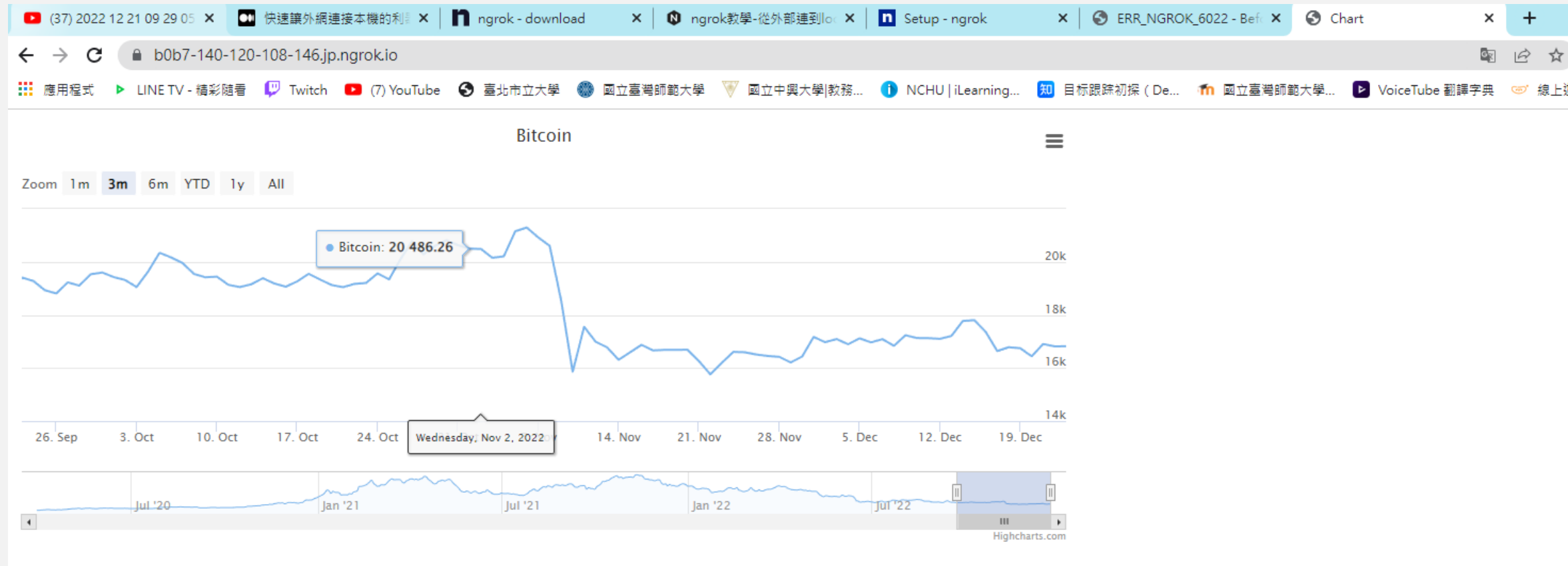
Connections         ttl    opn    rt1    rt5    p50    p90
                   0      0      0.00   0.00   0.00   0.00
```

在外網輸入網址



點visit site

Demo



GitHub(網址在備忘稿)

CiouQQ / NCHU_AIOT_HW5Public

<> Code

Issues

Pull requests

Actions

Projects

Wiki

Security

Insights

Settings

Pin

Unwatch1

Fork0

Star0

main1 branch0 tags

Go to file

Add file

<> Code

CiouQQ Update README.md624d50f now4 commits

aiotAdd files via upload1 minute ago

README.mdUpdate README.mdnow

README.md

NCHU_AIOT_HW5

NCHU_AIOT_HW5 將資料放到以下位置

↑ 本機 > 本機磁碟 (C:) > xampp > htdocs

名稱	修改日期	類型	大小
aiot	2022/12/23 下午 02:49	檔案資料夾	
dashboard	2022/12/12 下午 03:36	檔案資料夾	
img	2022/12/12 下午 03:36	檔案資料夾	
webalizer	2022/12/12 下午 03:36	檔案資料夾	
xampp	2022/12/12 下午 03:36	檔案資料夾	
applications	2022/6/16 上午 12:07	Chrome HTML D...	4 KB
bitnami	2022/6/16 上午 12:07	階層式樣式表文件	1 KB
favicon	2015/7/16 下午 11:32	圖示	31 KB
index	2015/7/16 下午 11:32	PHP 來源檔案	1 KB

About

NCHU_AIOT_HW5

Readme

0 stars

1 watching

0 forks

Releases

No releases published

Create a new release

Packages

No packages published

Publish your first package

Languages

Python 41.7%

HTML 34.2%

PHP 24.1%