

組別 3

資工三 4108056048 王宥嘉

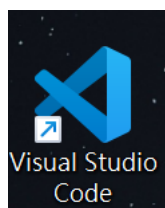
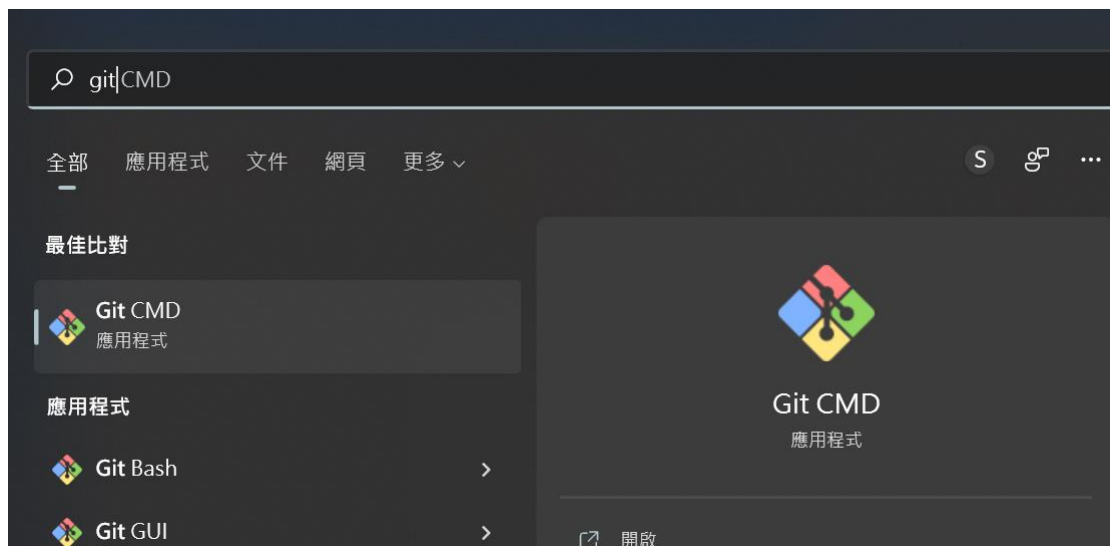
資工三 4108056051 鄭穎

資工三 4108056052 徐幸妤

Github 網址: https://github.com/HsuHsingYu/aiot_hw5

前置作業

1. 首先，在第一個步驟中，電腦需要先安裝下載好 vscode 和 git

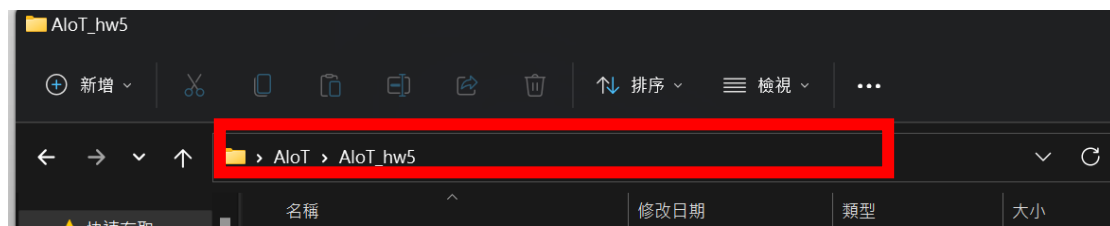


而 vscode 中需要安裝好此次作業所需的 python 相關模組，另外也需要去 python 官網下載相關的內容



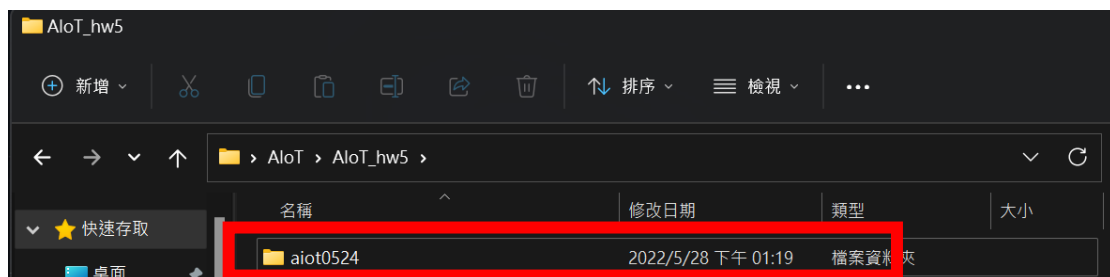
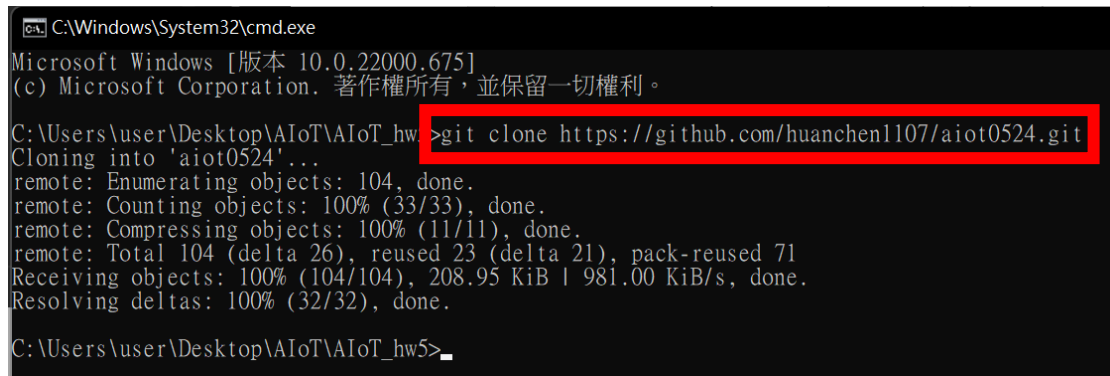


2. 接著我們需要建立一個資料夾，點擊下圖紅框區域輸入”cmd”來開啟終端機

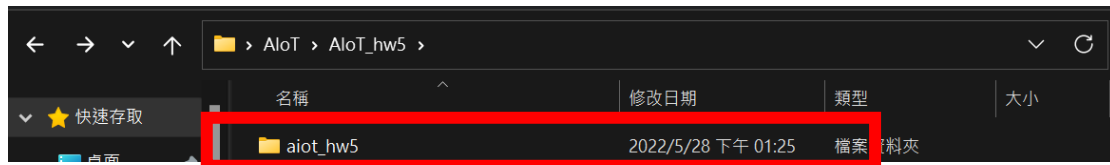


在裡面輸入 `git clone https://github.com/huanchen1107/aiot0524.git`

將老師上課所提供的 gitHub 內容抓下來



3. 然後再開一個資料夾，將 0524 中的 Readme.MD 抓過來，此為接下來主要作業的地方



建立在 github 中的 repository

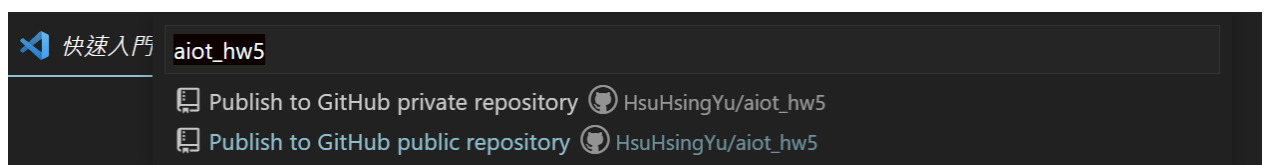
1. 將剛剛建立好的資料夾利用 vscode 開啟



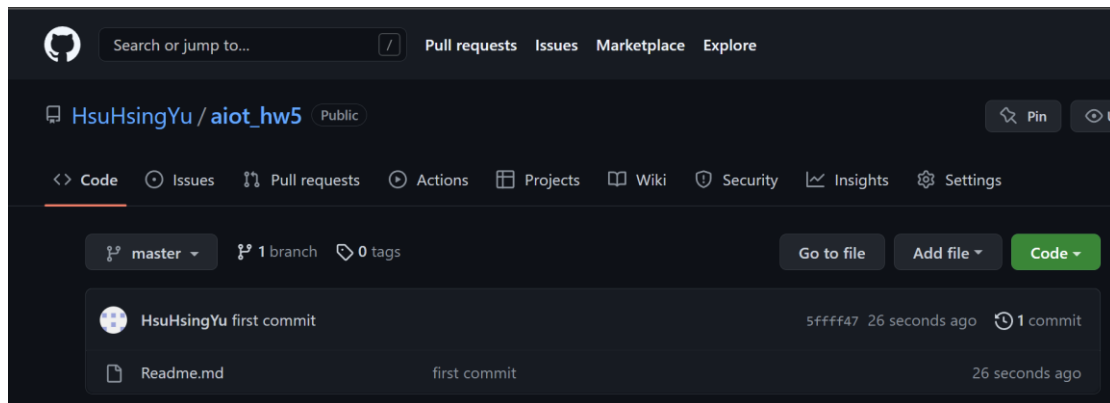
2. 點擊紅框，並選擇發布到 gitHub



3. 選擇 public 的



4. 發布成功



開始作業

1. 操作 check point1

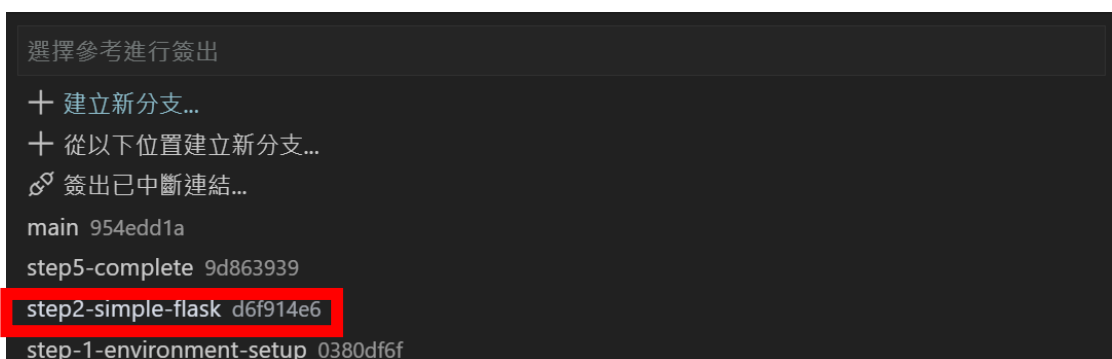
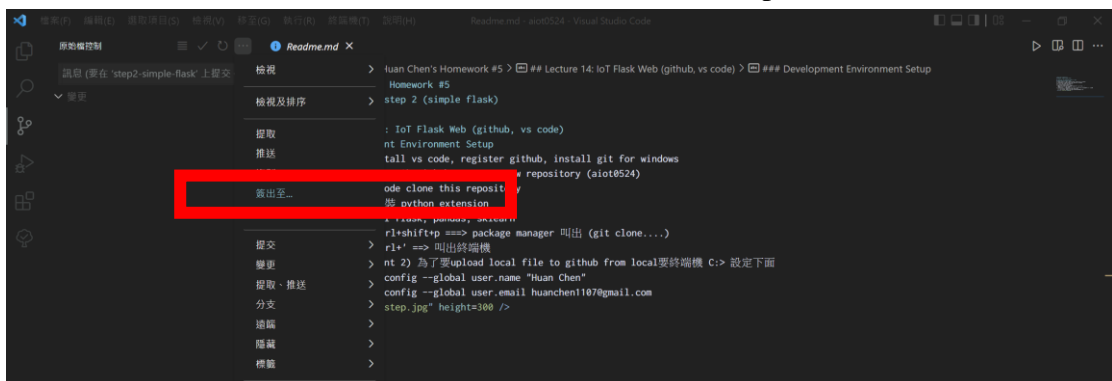
如前置作業所示(由於 flask、pandas、sklearn 等於課程前便已安裝，所以就不特別截圖了)

2. 操作 check point2

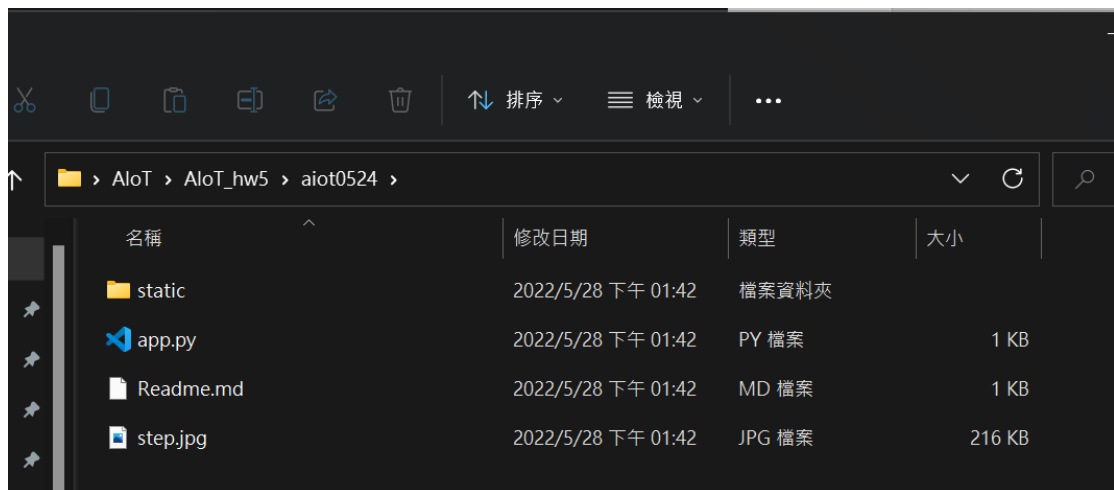
```
PS C:\Users\user\Desktop\AIoT\AIoT_hw5\aiot_hw5> git config --global user.name "HsuHsingYu"
PS C:\Users\user\Desktop\AIoT\AIoT_hw5\aiot_hw5> git config --global user.email s108056052@mail.nchu.edu.tw
```

3. 操作 step2

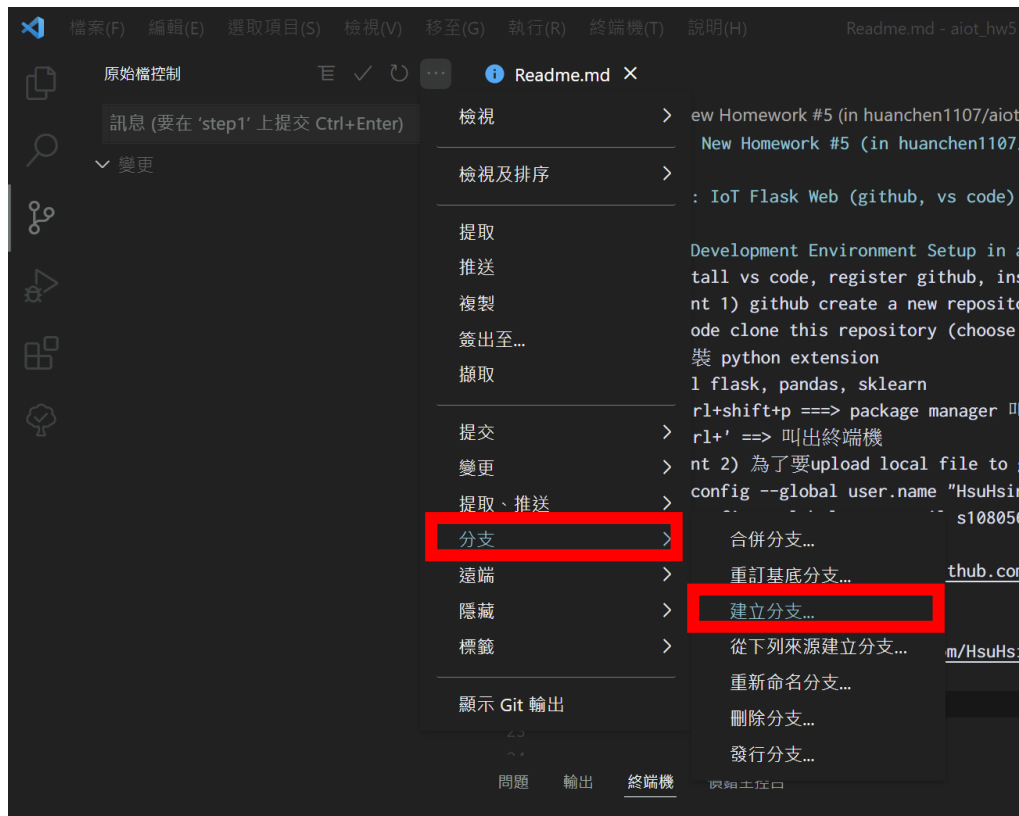
1. 使用 vscode 開始 ai0524 的資料夾，並簽出至 step2



完成後可以看到 aiot 資料夾的變化



2. 在 aiot_hw5 下面建立 step2 的分支(p.s step1 分支內容為將 readme 改為我們的資料後)



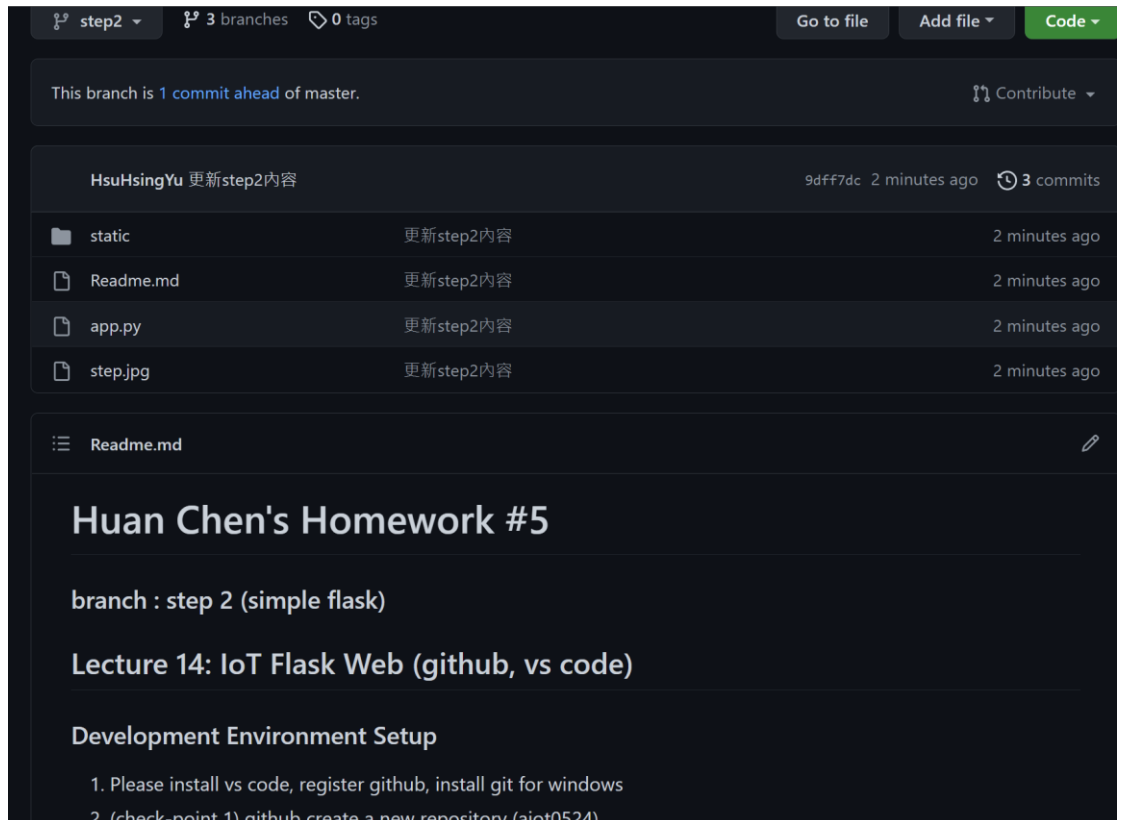
step2

請提供新的分支名稱 (按 'Enter' 鍵確認或按 'Esc' 鍵取消)

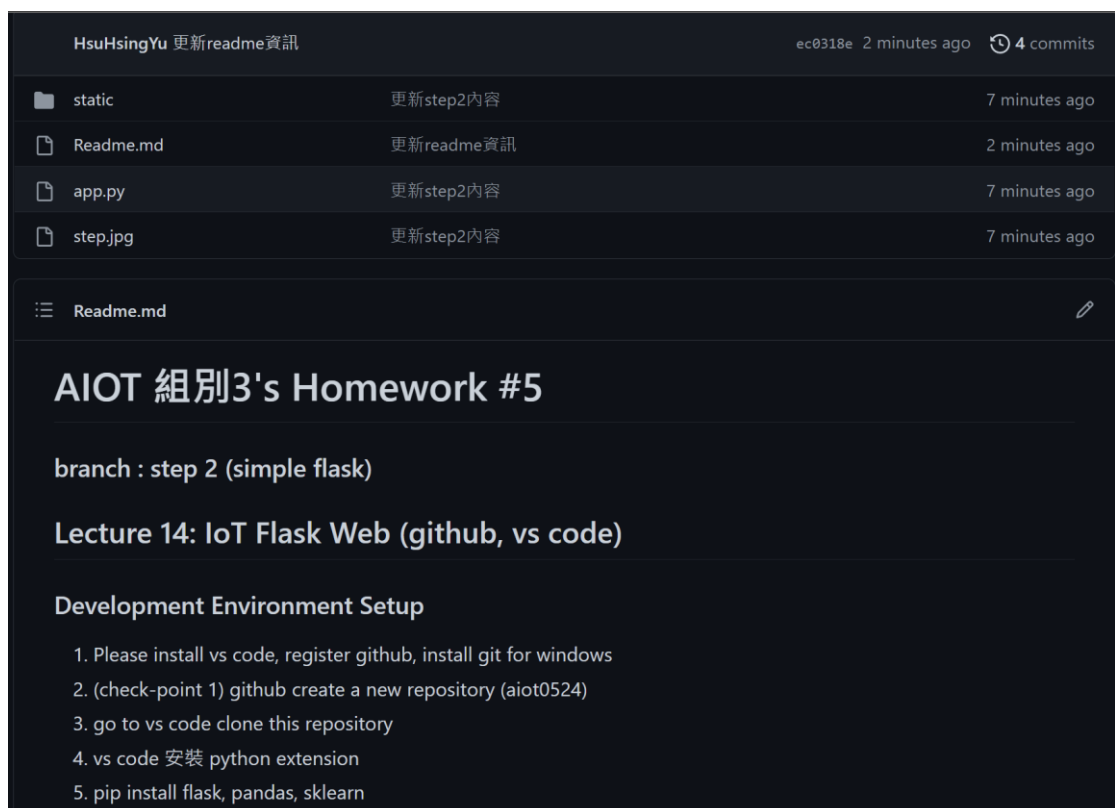
step2

可以由左下方看到目前已為 step2 分支

3. 將 aiot0524 目前的內容抓到 aiot_hw5 的資料夾中，並發布 step2 分支



4. 將內容改完我們的資訊後重新推上去



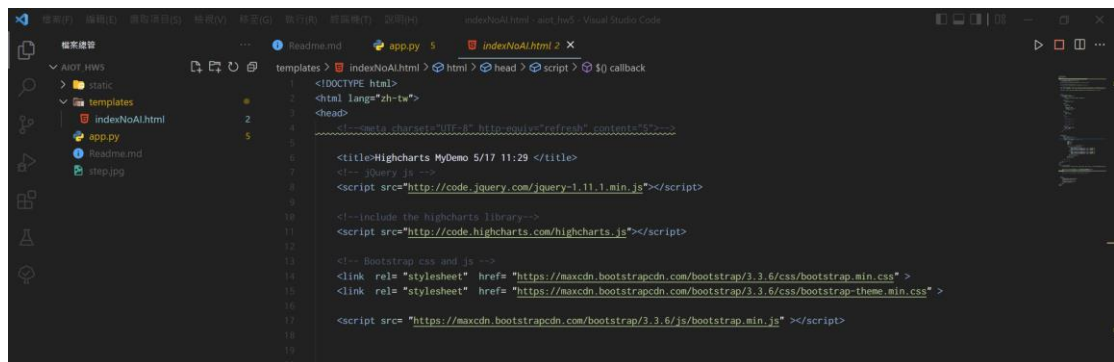
5. 操作 step3

在這個步驟中我們需要加入 indexNoAI.html 檔，並修改 app.py 中的內容
下圖為新加入檔案後的圖

1. 註冊網址 (“/noAI”)，進入 <http://127.0.0.1:5000/noAI> 後，會渲染到 indexNoAI.html 這個 template 上

```
28 @app.route("/noAI")
29 def noAI():
30     return render_template('indexNoAI.html')
```

2. 加入 indexNoAI.html



3. indexNoAI.html 有用到/setRandom 這個網址

```
69 $(function () {
70     $.ajax({
71         url: '/setRandom', //連接的URL
72         data: '{}', //來帶的參數
73         dataType: 'json', //資料格式
74         success: function(data) //傳送成功的function
75         {
76             lights = [];
77             humis=[];
78             temps = [];
79             time = [];
80
81             for (var i = 0; i < data.length; i++)
82             {
83                 if(parseInt(data[i][5])==0){
84                     lights.push({y:parseInt(data[i][2]), color: '#FF0000' });
85                     humis.push({y:parseInt(data[i][3]), color: '#FF0000' });
86                     temps.push({y:parseInt(data[i][4]), color: '#FF0000' });
87                 }else{
88                     lights.push({y:parseInt(data[i][2]), color: '#00FF00' });
89                     humis.push({y:parseInt(data[i][3]), color: '#00FF00' });
90                     temps.push({y:parseInt(data[i][4]), color: '#00FF00' });
91                 }
92                 time.push(data[i][1]);
93             }
94             highchartsinit();
95         } //success end
96     });
97 }
```

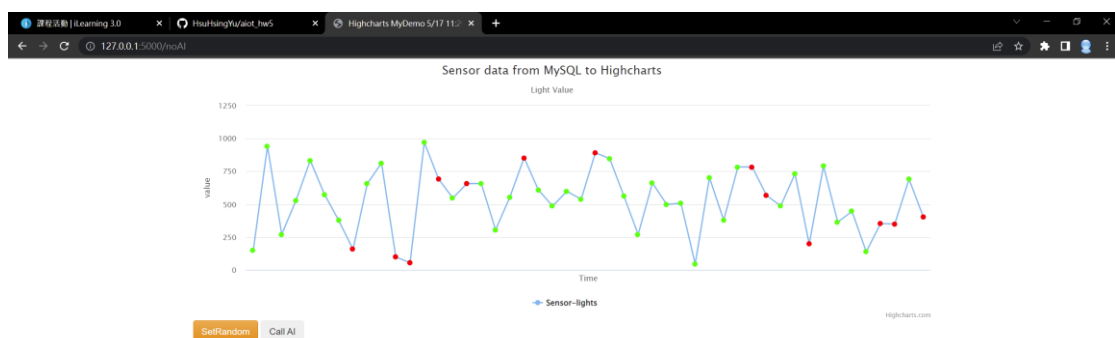
4. 在 app.py 裡註冊 “/setRandom” 和定義 getData function (導入到/setRandom 之後會做的事)

```

32 @app.route("/setRandom")
33 def getData():
34     myserver = "localhost"
35     myuser = "test123"
36     mypassword = "test123"
37     mydb = "aiotdb"
38
39     debug = 0
40     from pandas import DataFrame as df
41     import pandas as pd # 引用套件並縮寫為 pd
42     import numpy as np
43
44     import pymysql.cursors
45     #db = mysql.connector.connect(host="140.120.15.45", user="toto321", passwd="12345678", db="lightdb")
46     #conn = mysql.connector.connect(host=myserver, user=myuser, passwd=mypassword, db=mydb)
47     conn = pymysql.connect(host=myserver, user=myuser, passwd=mypassword, db=mydb)
48
49     c = conn.cursor()
50
51
52     #===== 執行 MySQL 查詢指令 =====#
53     c.execute("update sensors set value = RAND()*1000 where true")
54     conn.commit()
55
56     c.execute("SELECT * FROM sensors")
57
58     #===== 取回所有查詢結果 =====#
59     results = c.fetchall()
60     print(type(results))
61     print(results[:10])
62
63     if debug:
64         input("pause ....select ok.....")
65
66     test_df = df(list(results), columns=['id', 'time', 'value', 'temp', 'humi', 'status'])
67
68     print(test_df.head(10))
69     result = test_df.to_dict(orient='records')
70     seq = [[item['id'], item['time'], item['value'], item['temp'], item['humi'], item['status']] for item in result]
71     return jsonify(seq)
72     ##### cursor close, conn close
73     c.close()
74     conn.close()

```

5. 完成後執行 app.py 程式，在網址後面加上/noAI 即可看到此畫面



6. 操作 step4

在此步驟中我們會將 logistic regression 的 model 應用到作業中，亦會把程式與 db 互動的部分完成

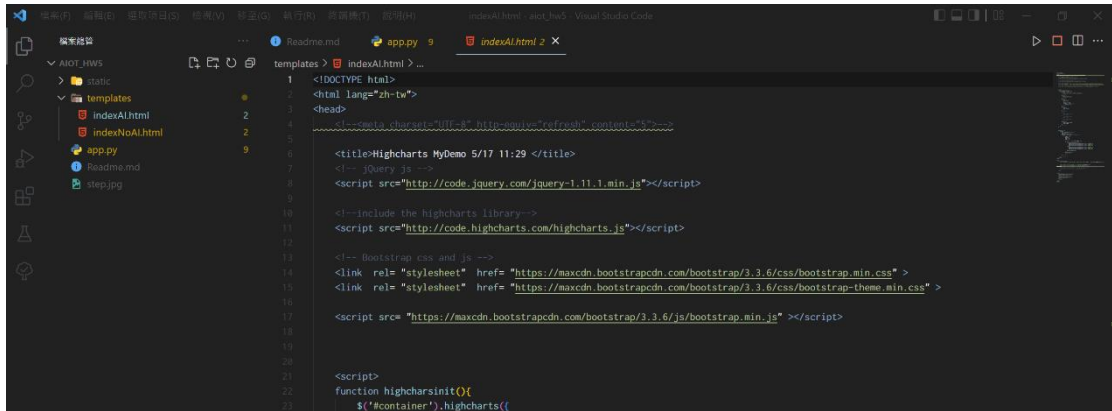
1. 註冊網址(“”), 進入 <http://127.0.0.1:5000> 後，會渲染到 indexAI.html 這個 template 上


```

24 @app.route("/")
25 def index():
26     return render_template('indexAI.html')

```

2. 加入 indexAI.html 檔



3. indexAI.html 裡有兩個按鈕，setRandom 和 callAI，按下 setRandom 後會到入到"/noAI"，按下 callAI 後會導入到"/"

```

117 <script>
118     $('#bSetRandom').click(function(){
119         window.location.href='/noAI';
120     });
121     $('#bCallAI').click(function(){
122         window.location.href='/';
123     });
124 </script>

```

4. indexAI.html 有用到/getPredict 這個網址

```

68     $(function () {
69         $.ajax({
70             url: '/getPredict',//連接的URL
71             data: "{}",//夾帶的參數
72             dataType: 'json', //資料格式
73             success: function(data) //傳送成功的function
74             {
75                 lights = [];
76                 humis=[];
77                 temps = [];
78                 time = [];
79
80                 for (var i = 0; i < data.length; i++)
81                 {
82                     if(parseInt(data[i][5])==0){
83                         lights.push({y:parseInt(data[i][2]), color: '#FF0000' });
84                         humis.push({y:parseInt(data[i][3]), color: '#FF0000' });
85                         temps.push({y:parseInt(data[i][4]), color: '#FF0000' });
86                     }else{
87                         lights.push({y:parseInt(data[i][2]), color: '#00FF00' });
88                         humis.push({y:parseInt(data[i][3]), color: '#00FF00' });
89                         temps.push({y:parseInt(data[i][4]), color: '#00FF00' });
90                     }
91                     time.push(data[i][1]);
92                 }
93                 highchartsinit();
94             } //success end
95
96         }); //ajax end

```

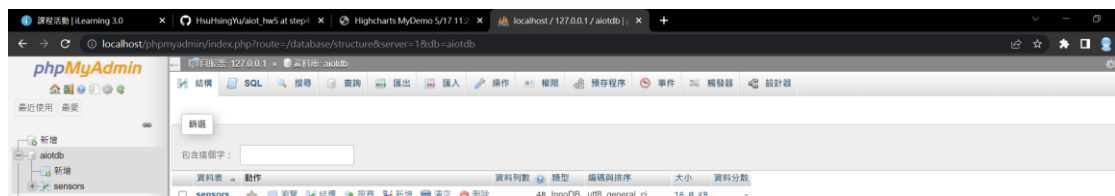
5. 在 app.py 裡註冊"/getPredict"

```

76 @app.route("/getPredict")

```

6. 匯入 database 的相關資訊



6. 定義 getPredict function(導入到/setRandom 之後會做的事)

(1) 設定變數和引用套件

```

77 def getPredict():
78     #==== step 1: setup variable =====
79     myserver ="localhost"
80     myuser="test123"
81     mypassword="test123"
82     mydb="aiotdb"
83
84     debug =0
85     from pandas import DataFrame as df
86     import pandas as pd # 引用套件並縮寫為 pd
87     import numpy as np

```

(2) 讀取 model

```

89     #step 2: load model  #讀取Model###
90     import pickle
91     import gzip
92     with gzip.open('./model/myModel.pgz', 'r') as f:
93         model = pickle.load(f)

```

(3) 從資料庫裡拿到測試資料

```

95     # step 3: get test data from database
96
97
98     import pymysql.cursors
99     #db = mysql.connector.connect(host="140.120.15.45",user="toto321", passwd="12345678", db="lightdb")
100    #conn = mysql.connector.connect(host=myserver,user=myuser, passwd=mypassword, db=mydb)
101    conn = pymysql.connect(host=myserver,user=myuser, passwd=mypassword, db=mydb)
102
103    c = conn.cursor()
104    if debug:
105        input("pause.. conn.cursor() ok.....")
106
107    #===== 執行 MySQL 查詢指令 =====#
108    c.execute("SELECT * FROM sensors")
109
110    #===== 取回所有查詢結果 =====#
111    results = c.fetchall()
112    print(type(results))
113    print(results[:10])
114    if debug:
115        input("pause ....select ok.....")
116
117    test_df = df(list(results),columns=['id','time','value','temp','humi','status'])
118
119    print(test_df.head(10))

```

(4) 用 model 預測

```

122    testX=test_df['value'].values.reshape(-1,1)
123    testY=model.predict(testX)
124    print(model.score(testX,testY))
125
126    test_df['status']=testY
127    print(test_df.head(10))
128
129    if debug:
130        input("pause.. now show correct one above.....")

```

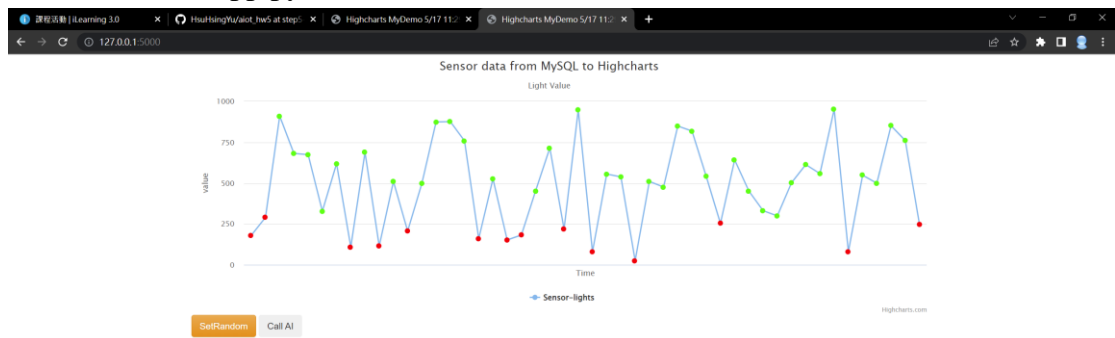
(5) 寫回資料庫

```

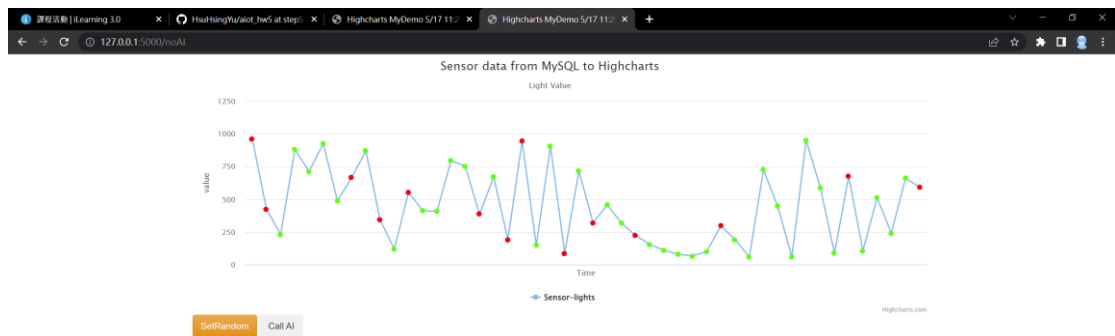
148    ##Example 2 ## write back mysql #####
149    ## make all status =0
150    c.execute('update sensors set status=0 where value>0')
151
152    ## choose status ==1 have their id available
153    id_list=list(test_df[test_df['status']==1].id)
154    print(id_list)
155
156    for _id in id_list:
157        #print('update light set status=1 where id='+str(_id))
158        c.execute('update sensors set status=1 where id='+str(_id))
159
160    conn.commit()
161
162
163    result = test_df.to_dict(orient='records')
164    seq = [[item['id'], item['time'], item['value'], item['temp'], item['humi'], item['status']] for item in result]
165    return jsonify(seq)

```

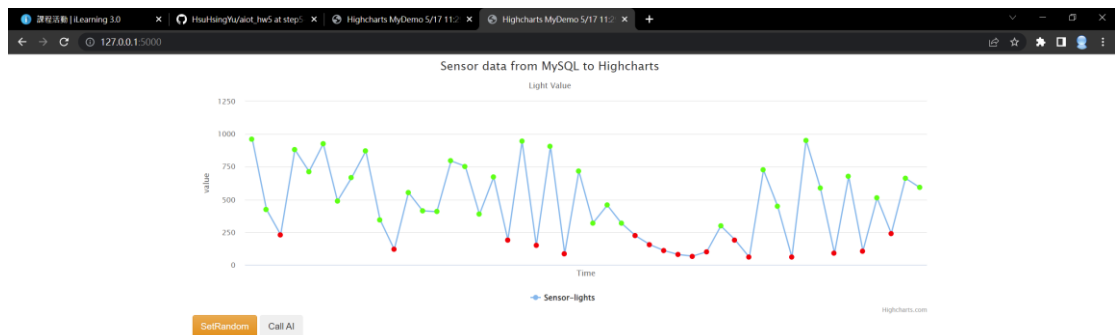
7. 完成後執行 app.py 程式，即可看到此畫面



8. 按下 SetRandom 之後



9. 按下 CallAI 之後



7. 操作 step5

完成，將 ai0t0524 中的內容抓至目前資料夾中，並更新相對應的資訊

課程活動 | Learning 3.0

HsuHsingYu/aiot_hw5_at_step5

Highcharts MyDemo 5/17 11:00

Highcharts MyDemo 5/17 11:00

github.com/HsuHsingYu/aiot_hw5/tree/step5-complete

HsuHsingYu update67554e7 4 minutes ago13 commits

db加入model16 minutes ago

model加入model16 minutes ago

templatesupdate5 minutes ago

Readme.mdupdate4 minutes ago

app.pyupdate5 minutes ago

Readme.md

AIOT 組別3's Homework #5

Lecture 14: IoT Flask Web (github, vs code)

Development Environment Setup

1. Please install vs code, register github, install git for windows
2. (check-point 1) github create a new repository (aiot0524)
3. go to vs code clone this repository
4. vs code 安裝 python extension
5. pip install flask, pandas, sklearn

- 快捷鍵 ctrl+shift+p ==> package manager 叫出 (git clone...)
- 快捷鍵 ctrl+shift+p ==> 叫出校課程

No packages published

Publish your first package