

推送廣告活動訊息

學習目標：

- 利用已學過的知識，進行商業化活動！

建立廣告活動訊息資料庫

1. 增加一個資料表 ads 存放訊息：

```
C:\workspace\LineBot> heroku login -i
C:\workspace\LineBot> python
>>> import os
>>> import psycopg2
>>> heroku_pgCLI = 'heroku config:get DATABASE_URL -a 你的APP名稱'
>>> DATABASE_URL = os.popen(heroku_pgCLI).read()[:-1]
>>> print(DATABASE_URL)
>>> connection_db=psycopg2.connect(DATABASE_URL, sslmode='require')
>>> cursor = connection_db.cursor()
>>> SQL_cmd = '''CREATE TABLE ads(
... ads_id serial PRIMARY KEY,
... adstitle VARCHAR (150) NOT NULL,
... adscontent TEXT NOT NULL,
... adspicname VARCHAR (150) NOT NULL
... );'''
>>> cursor.execute(SQL_cmd)
>>> connection_db.commit()
>>> cursor.close()
>>> connection_db.close()
>>>
```

2. 修改 app/dataSQL/connectDB.py，寫入一段顯示活動資料的SQL語法：

```
def showAds():
    DATABASE_URL = os.environ['DATABASE_URL']
    connection_db = psycopg2.connect(DATABASE_URL,sslmode='require')
    cursor = connection_db.cursor()
    SQL_cmd = f"""SELECT * FROM ads ORDER by ads_id DESC limit 5;"""
    cursor.execute(SQL_cmd)
    connection_db.commit()
    rows = cursor.fetchall()
    data = []
    for raw in rows:
        data.append(raw)

    cursor.close()
    connection_db.close()
    return data
```

3. 修改 app/router.py 檔案，加入一段引導到活動列表的程式：

```
# 顯示活動清單列表
@app.route("/showads")
def showads():
    adslists = connectDB.showAds()
    return render_template('showads.html',menulist=adslists)
```

4. 增加一個顯示活動清單的 html 程式碼 app/templates/showads.html：

```

{% extends "base.html" %}
{% block title %}大學麵店活動通知系統{% endblock %}

{% block main %}

<!-- 宣告巨集 -->
{% macro show_row(data, tag) -%}
{% for menu in data %}
<div class="row">
    {% if menu[4] == 'Action' %}
        {% for item in menu %}
            <div class="col">
                <{{ tag }}>{{ item }}</{{ tag }}>
            </div>
        {% endfor %}
    {% else %}
        {% for item in menu %}
            <div class="col">
                <{{ tag }}>{{ item }}</{{ tag }}>
            </div>
        {% endfor %}
        <div class="col">
            <button type="submit" value="{{ menu[0],menu[1] }}" name="deldata">刪除</button>
        </div>
    {% endif %}
</div>
{% endfor %}
{%- endmacro %}
<!-- 宣告結束 -->

<div class="container my-5 py-5">
    <div class="row">
        <div class="col">
            <h1>活動列表</h1>
        </div>
    </div>
    <!-- 顯示菜單用 -->
    {% set cols = (("項次", "活動標題", "活動內容", "參考圖檔", "Action"),)%}

    {{ show_row(cols, "h2") }}
    <hr class="my-4">
    <form action="/delmenu" method="post">
        {{ show_row(menulist, "p") }}
    </form>
    <form action="/addmenu" method="get">
        <input type="submit" value="新增活動">
    </form>
</div>

{% endblock %}

```

5. 推送上 Heroku 專案！檢視一下成果！

6. 修改 app/templates/showads.html 檔案內容，確定新增/刪改的路徑：

```

{{ show_row(cols, "h2") }}
<hr class="my-4">
<form action="/delads" method="post">
    {{ show_row(menulist, "p") }}
</form>
<form action="/addads" method="get">
    <input type="submit" value="新增活動">
</form>

```

7. 修改 app/router.py 檔案內容，加上新增與刪除的程式碼：

```

# 新增活動清單
@app.route("/addads", methods = ['GET', 'POST'])
def addads():
    if request.method == 'POST':
        return showads()
    else:
        return showads()
# 刪除活動清單
@app.route("/delads", methods = ['GET', 'POST'])
def delads():
    if request.method == 'POST':
        print(request.form)
        data = ""
        data1 = ""
        for key, value in request.form.items():
            data = eval(''.join(value))
            data1 = str(data[0])+' '+str(data[1])
            print(data1)
            connectDB.deleteAds(data1)
        return showads()
    else:
        return render_template("showads.html")

```

8. 修改 app/dataSQL/connectDB.py 程式，加入下列程式碼：

```

(前面略過....)
def deleteAds(text):
    DATABASE_URL = os.environ['DATABASE_URL']
    connection_db = psycopg2.connect(DATABASE_URL, sslmode='require')
    cursor = connection_db.cursor()
    data = text.split(',')
    SQL_cmd = f"DELETE FROM ads WHERE ads_id = %s;"
    print(int(data[0]))
    cursor.execute(SQL_cmd, [int(data[0])])
    connection_db.commit()
    cursor.close()
    connection_db.close()
    return text

```

9. 推送上 Heroku 專案！檢視一下成果！

10. 增加一個填寫資料內容的檔案 app/templates/addads.html：

```

{% extends "base.html" %}
{% block title %}大學麵店活動訊息系統{% endblock %}
{% block main %}
<div class="container">
    <div class="row">
        <div class="col">
            <h1>新增活動訊息</h1>
            <form action="/addads" method="post">
                <label for="adstitle">訊息標題</label>
                <input type="text" id="adstitle" placeholder="請輸入訊息標題" name="adstitle">
                <br>
                <label for="adscontent">訊息內容</label>
                <input type="text" id="adscontent" placeholder="請輸入訊息內容" name="adscontent">
                <br>
                <label for="adspicname">圖片名稱</label>
                <input type="text" id="adspicname" placeholder="請輸入圖片名稱" name="adspicname">
                <br>
                <input type="submit" value="送出">
            </form>

```

```

        </div>
    </div>
</div>
{% endblock %}

```

11. 修改檔案 app/timer.py :

```

(前面略過....)
@scheduleEvent.scheduled_job('interval',minutes=1)
def timed_job():
    msg = connectDB.showAds()
    print(msg)
    line_bot_api.push_message("你的 I D",TextSendMessage(text='test'))

```

12. 修改檔案 app/dataSQL/connectDB.py :

```

(前面略過....)
def addads(text):
    DATABASE_URL = os.environ['DATABASE_URL']
    connection_db = psycopg2.connect(DATABASE_URL,sslmode='require')
    cursor = connection_db.cursor()
    data = text.split(',')
    table_columns = '(adsname,adscontent,adspicname)'
    SQL_cmd = f"INSERT INTO menu { table_columns } VALUES(%s,%s,%s);"
    cursor.execute(SQL_cmd,(str(data[0]),str(data[1]),str(data[2])))
    connection_db.commit()
    cursor.close()
    connection_db.close()
    return text

```

13. 修改 add/router.py 檔案內容 :

```

(前面略過....)
# 新增活動清單
@app.route("/addads",methods = ['GET','POST'])
def addads():
    if request.method == 'POST':
        print(request.form)
        data = ""
        for key,value in request.form.items():
            data += value
            data += ','
        data = data[:-1]
        connectDB.addads(data)
        return showads()
    else:
        return render_template("addads.html")
(後面略過....)

```

14. 推送上 Heroku 專案 !