

## 使用 Flask Login 模組

學習目標：

- 利用 Flask Login 模組，控制站台運作！

### 使用 Flask Login 模組

1. 新增 app/templates/login.html 檔案：

```
{% extends "base.html" %}
{% block title %}大學麵店系統管理區{% endblock %}

{% block main %}
<div class="container">
  <div class="row">
    <div class="col">
      <h1>後台管理</h1>
      <form action="/logins" method="post">
        <label for="loginname">登入帳號</label>
        <input type="text" id="loginname" placeholder="請輸入帳號" name="loginname">
        <br>
        <label for="password">登入密碼</label>
        <input type="password" id="password" placeholder="請輸入密碼" name="password">
        <br>
        <input type="submit" value="送出">
      </form>
    </div>
  </div>
{% endblock %}
```

2. 修改 requirements.txt

```
(前面略過....)
Flask-Login>=0.4.1
```

3. 修改 app/\_\_init\_\_.py

```
(前面略過....)
from flask_login import LoginManager
(中間略過....)
# 導入加密 key
app.secret_key = config.get('Flask', 'SecretKey')

lm = LoginManager()
lm.init_app(app)
lm.session_protection = "strong"
lm.login_view = 'login'
lm.login_message = '請輸入帳號密碼'

# 導入其他的程式模組
from app import router, linebotmodules, loginmodels
```

4. 使用 terminal 介面，產生 secret\_key 內容：

```
C:\workspace\LineBot> python
>>> import os
>>> os.urandom(16).hex()
(產生一堆編碼)
```

5. 修改 config.ini 檔案內容：

```
[Flask]
SecretKey = 上個步驟產生的一堆編碼
```

#### 6. 新增 app/loginmodels.py 檔案內容：

```
from flask_login import UserMixin
from app import lm

class User(UserMixin):
    pass

@lm.user_loader
def user_loader(users):
    if users not in users_dict:
        return

    user = User()
    user.id = users
    return user

@lm.request_loader
def request_loader(request):
    user = request.form.get('user_id')
    if user not in users_dict:
        return

    user = User()
    user.id = user

    user.is_authenticated = request.form['password'] == users_dict[user]['password']

    return user

users_dict = {'Owner': {'password': 'HelloWorld'}}
```

#### 7. 修改路由檔案 app/router.py

```
(前面略過....)
from flask import request, render_template, url_for, redirect, flash
from flask_login import login_user, logout_user, login_required, current_user
from app.loginmodels import User, users_dict

@app.route('/logins', methods=['GET', 'POST'])
def login():
    if request.method == 'GET':
        return render_template("login.html")

    user = request.form['loginname']
    conditionA = user in users_dict
    conditionB = request.form['password'] == users_dict[user]['password']

    if conditionA and conditionB:
        user1 = User()
        user1.id = user
        login_user(user1)
        flash(f'{user}!Welcome home!')
        return redirect(url_for('home'))

    flash('login failed...')
    return render_template('login.html')

@app.route('/logout')
def logout():
    user = current_user.get_id()
    logout_user()
    flash(f'{user}!logout!')
    return render_template('login.html')

@app.route('/show_records')
@login_required
def show_records():
```

```
python_records =web_select_overall()
return render_template('show_records.html', html_records=python_records)
(後面略過....)
```

#### 8. 修改 app/templates/base.html

```
{% if current_user.is_authenticated %}
    <li class="nav-item">
        <a class="nav-link" href="http://mylinebothellotux.herokuapp.com/showmenu">麵店菜單</a>
    </li>
    <li class="nav-item">
        <a class="nav-link" href="http://mylinebothellotux.herokuapp.com/showads">活動訊息</a>
    </li>
    <li class="nav-item">
        <a class="nav-link" href="http://mylinebothellotux.herokuapp.com/logout">登出</a>
    </li>
{% else %}
    <li class="nav-item">
        <a class="nav-link" href="http://mylinebothellotux.herokuapp.com/logins">登入</a>
    </li>
{% endif %}
```

#### 9. 將專案送上 Heroku 主機！

#### 10.