**崑山科技大學**

**資訊工程系**

**學生專題製作報告**

**CNN模型判別肺炎X光影像Web平台系統**

**A Web Platform System of CNN Model Discriminate Lungs’ X-ray Image**

**指導教授：李宗儒**

**專題組員1：蘇宇祥 學號：**4070E021

**專題組員2：陳冠喨 學號：**4070E020

**專題組員3：沈明楷 學號：**4070E022

**中華民國 111年 05 月**

**CNN模型判別肺炎X光影像Web平台系統**

蘇宇祥 陳冠喨 沈明楷

崑山科技大學資訊工程系

摘 要

新型冠狀肺炎（Covid-19, Corona Virus Disease 2019）近年來肆虐全球，台灣在2020年初就出現第一起病例，然而新型冠狀肺炎症狀有發燒、咳嗽、肌肉痠痛、嗅味覺失調等症狀，更是會讓肺部產生纖維化，因此當醫護人員檢測時，除了進行抗原快篩與核酸檢測（PCR）外，醫師會透過照胸腔X光來查看病情，不過隨著病例增加，以及利用傳統的人工判讀肺部X光影像檢驗方式，醫師查看相關資料時可能會有誤判情況發生，可能導致患者失去黃金醫療時間，為此希望能夠解決判讀上的問題。

而現在科技日新月異，漸漸地人工智慧（Artificial Intelligence, AI）與影像辨識技術開始與醫療結合，渴望透過科技解決當前所遇問題，達到AI生物醫療，希望可以藉由人工智慧協助，讓醫師有更精確的判斷，患者能夠及早治療。

**誌 謝**

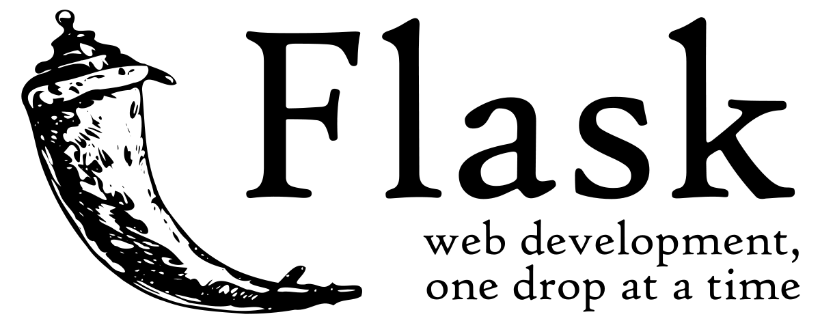
在大學專題製作期間，作品與報告書能完成，需要感謝本組指導教授李宗儒老師，在此獻上最大的謝意。專題一開始李宗儒老師就慢慢引導我們，讓我們思考想要做什麼類型的專題，解決哪種狀況等，在製作過程中，也誨人不惓的耐心指導與協助，提供良好的學習環境與資源，更是在我們遇到瓶頸時，給予我們適當的引導。

從當初的摸索到對專題有想法後再實作出來，這段過程當中，雖然我們遇到了許多的障礙，很多時候都讓我們不知自己的方向該如何做，同時也以目前的能力與上課中所學的知識來製作此專題，因應老師所說，專題做完不代表結束，這只是表面的結束，重要的是從中學習到了甚麼，例如適時的報告進度（時間的控管）、與一個上進的態度（愉快的學習）。看看自己所製作的專題，自己也是知道還不夠完美，也沒達到理想的標準，只有不斷精進的能力，才有更好更完美地作品。

**專題架構圖**

因應現今肺炎疫情蔓延，醫療院所為檢疫與掌握病患是否罹患肺炎，過去的技術是由醫師將患者肺部X光與罹患肺炎X光，以肉眼比對，往往會造成誤判，很可能導致患者失去黃金醫療時間，本組藉由人工智慧協助，醫師有更精確的判斷，患者能夠及早治療。

**Flask介紹**

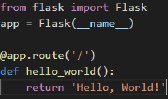
****

Flask是Python的輕量web微框架，是由Jinja2模板引擎和Werkzeug WSGI(Web Server Gateway Interface)工具箱組成，透過flask可以輕易架設網站，使用CMD(命令提示字元)pip來安裝Flask套件。pip 是Python 安裝管理套件工具，所以我們需要用到pip install flask來進行下載。



**(CMD命令提示字元安裝flask)**

**Hello World(app.py)**



**Route(路由)**

使用者可透過自己所定義的路由，來訪問網頁。程式中的@app.route("/")是路由的位置(/代表預設首頁)，CMD(命令提示字元)輸入flask run來執行，預設網址localhost (<http://127.0.0.1:5000/>)，假設網站是/flask，那@app.route("/flask")即可建立網路路由。

**HTTP Methods(GET & POST requests)**

GET跟POST requests可透過@app.route()新增method來處理HTTP請求。

* Request:是用來取得數據，需透過from flask import request來import套件。
* GET:將數據傳送，利用『？』作為第一個參數的連接符號，第二個之後的參數是以『&』符號作為連接符號。
* POST:由於使用GET方法傳送資料會曝露資訊在網址列，若要使用安全性較高的方式就是使用POST資料表單發送到伺服器。

**templates(動態文件)**

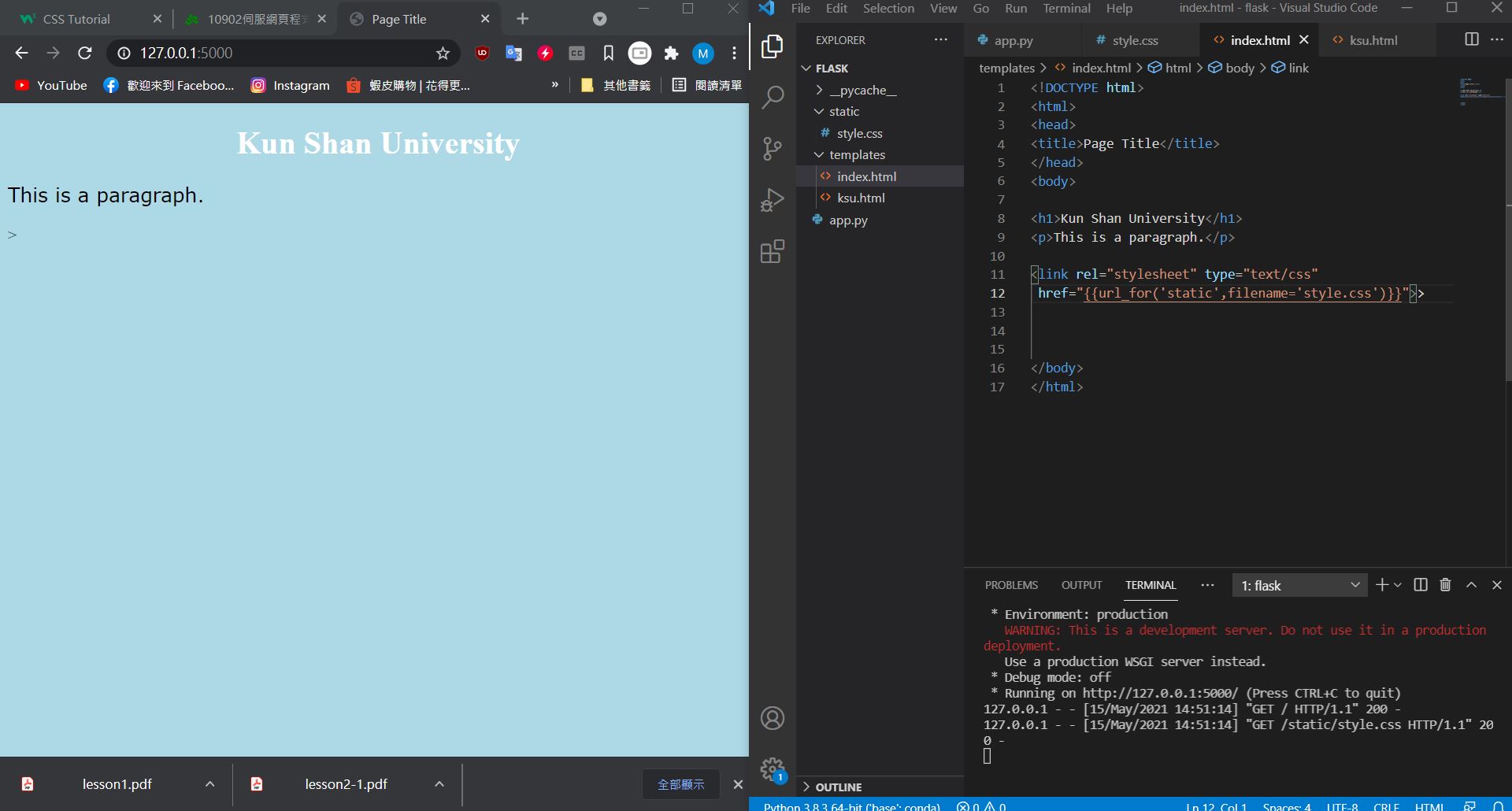
專案文件夾中新增templates資料夾，可以將副檔名.html的檔案放入，利用render\_template()渲染模板可以將html檔進行渲染。

**Static(靜態文件)**

若flask需要處理static靜態檔案(CSS、Javascript、圖片)，需要在該專案資料夾中新增static資料夾，將欲加入的靜態檔案放入static資料夾中才可以使用。

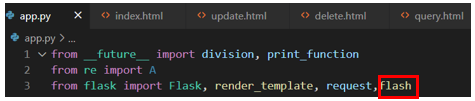
**url\_for()、redirect**

透過url\_for取得函數的URL，並非是route路徑。而redirect可以讓flask重新導向頁面，利用url\_for與redirect的搭配我們就可以重新導向指定的頁面。

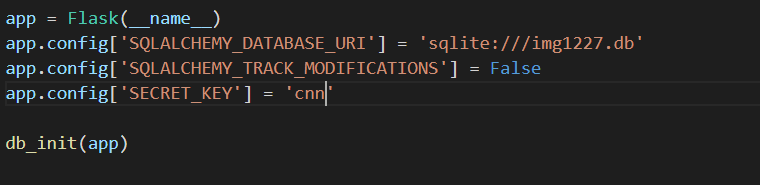


**flash message**

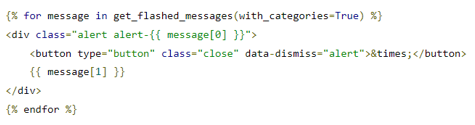
當使用者提交資料後，若資料的格式不符合或是要告訴使用者處理流程是否完成，可以透過flash訊息顯示的方式告訴使用者哪個資料填寫錯誤不符合，或者是使用者登入成功，會跳出message來告訴使用者，首先使用者需要先import flash，flask有用到session需要一組密碼進行secret key加密，透過迴圈來取得訊息，來判斷有無message訊息，例如:本專題的刪除功能，若使用者成功刪除的該筆紀錄，則會顯示綠色的該筆紀錄已刪除字樣。



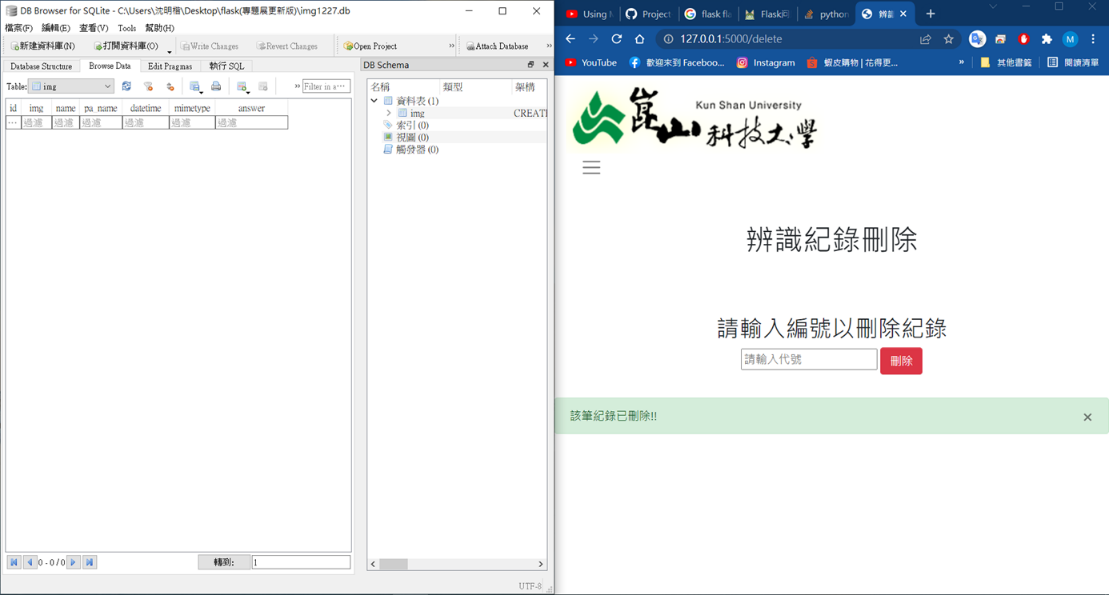
(import flash)



(設定secret key加密)



(將迴圈程式新增到欲顯示的html)



(刪除介面的flash message)

**extands(模板繼承)**

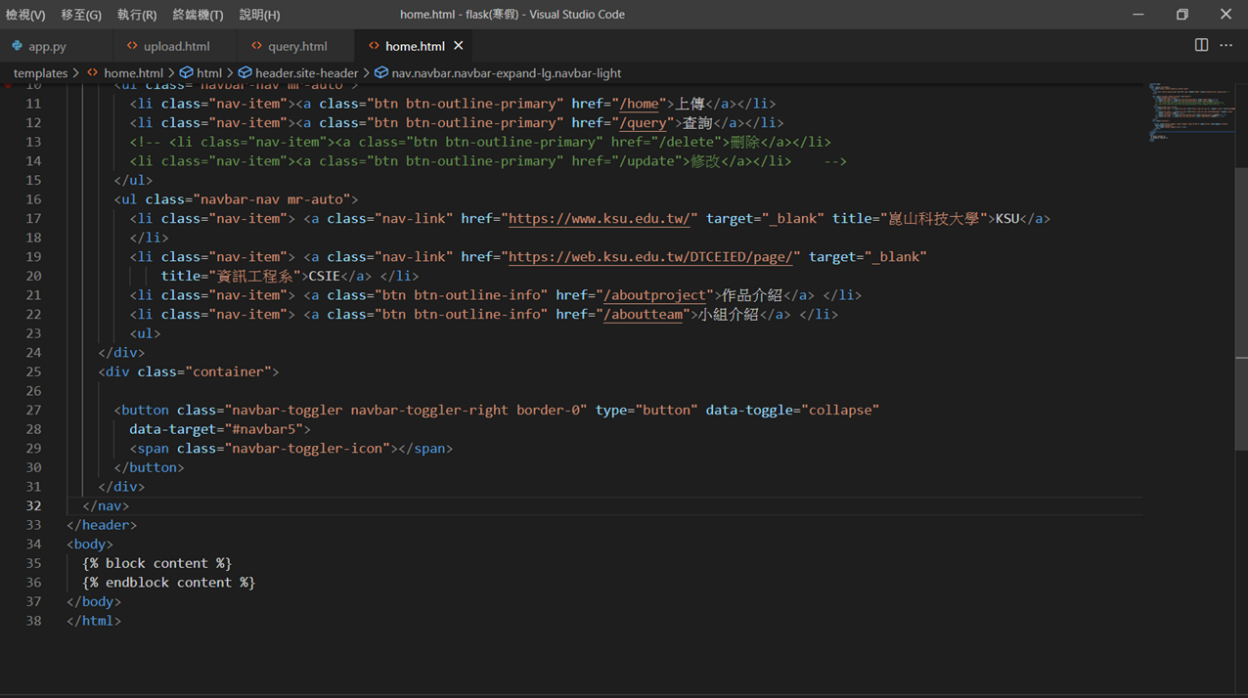
利用{% block content %}{% endblock %}放父樣板所沒有的內容。使用block區塊是子樣板的關鍵字，content是子模板本體的名稱。在同一個頁面content的名稱不能一樣。



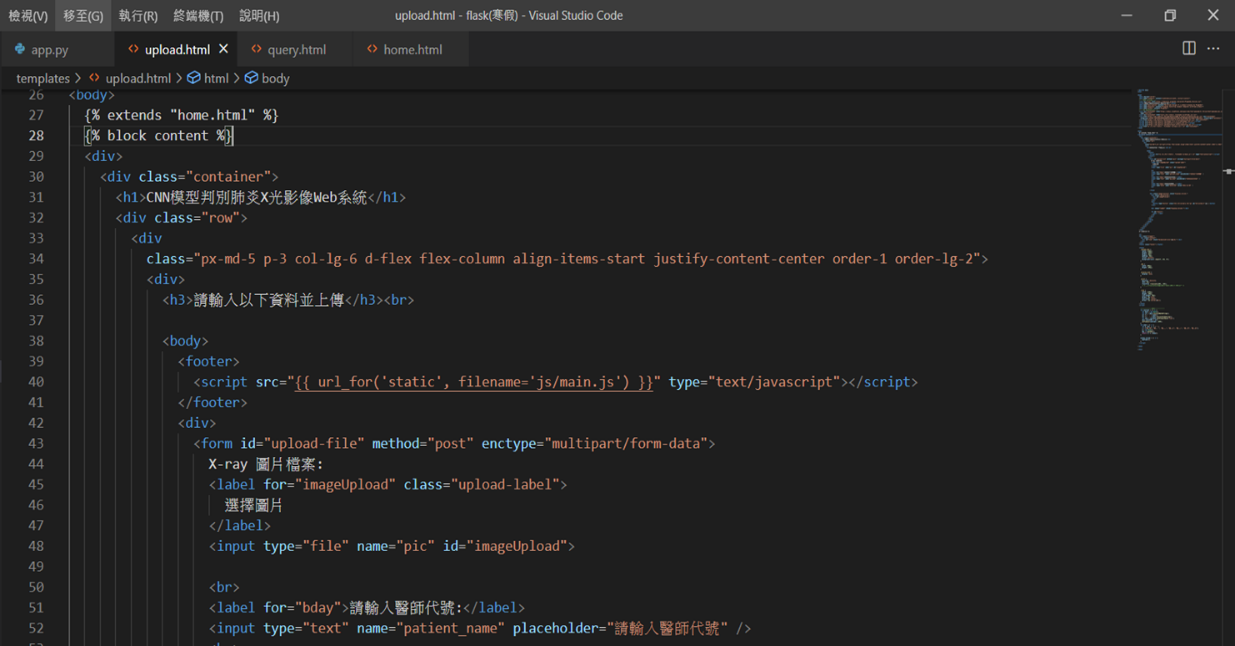
(紅框處自定義可做切換 <https://www.itread01.com/content/1546835479.html>)

例如:

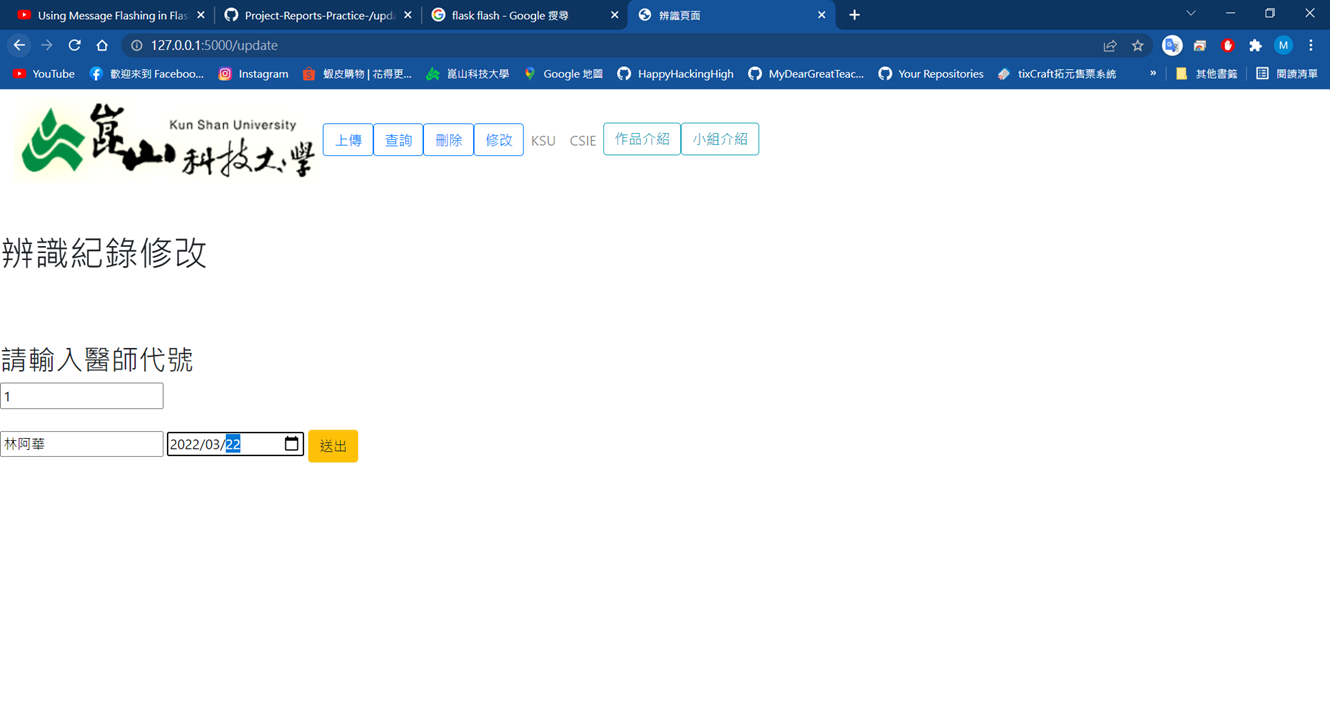
本專題的home.html需要給子模板(upload.html)繼承(extands) 的導覽列navbar，home.html的{% block block content %}{% endblock content %}要接upload.html的{% extends "base.html" %}{% block content%}{% endblock content%}的範圍。



(home.html父模板)



(upload.html子模板)



(upload.html已繼承home.html的navbar導覽列)

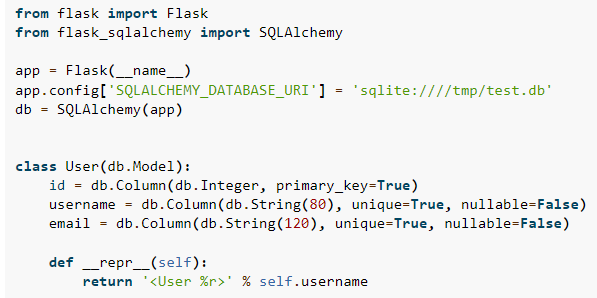
總結:

1. 有點像在block的地方預留一個孔洞，子模板就可以套用這個樣板的頁面。
2. 如果navbar需要做修改，只需要更改home.html(父模板)，重新上傳之後全部只要有extends的子模板就會繼承父模板了。

**Flask SQLAlchemy**

Flask本身不支援SQL操作，SQLAlchemy支援物件關係對映器

ORM(Object Relational Mapper)框架來使用資料庫，SQLAlchemy可使用SQLite、PostgreSQL、MySQL、Oracle，使用ORM 來操作資料庫，透過pip install flask-sqlalchemy來安裝套件。



<https://flask-sqlalchemy.palletsprojects.com/en/2.x/quickstart/>

使用 app.config 來連線到database設定SQLite資料庫的路徑，本專題使用較輕量的資料庫SQLite進行資料庫的操作，下方class去定義資料庫欄位，如上圖所示，這是一個資料表為User的table(表格)id、username、email 為資料庫欄位(Column)。

* db string的格式(Integer、float、string……等)
* unique = True (設定為唯一性)
* index = True (索引值)
* nullable = True (允許為null空值)
* default = ‘預設值’

|  |  |  |
| --- | --- | --- |
| **id** | **name** | **age** |
| **1** | **Ksu** | **18** |
| **2** | **Steven** | **22** |
| **3** | **Grady** | **21** |

**初始化資料庫**

from model import db

db.create\_all()

**新增(insert)**

from import User

user = User(‘Ksu’,’18')

db.session.add(user)

db.session.commit()

**修改(update)**

user = User.query.filter\_by(name=’Ksu’).first()

user.age=22

db.session.commit()

**刪除(delete)**

user = User.query.filter\_by(name=’Ksu’).first()

db.session.delete(user)

db.session.commit()

**query(查詢)filter&filter\_by過濾器**

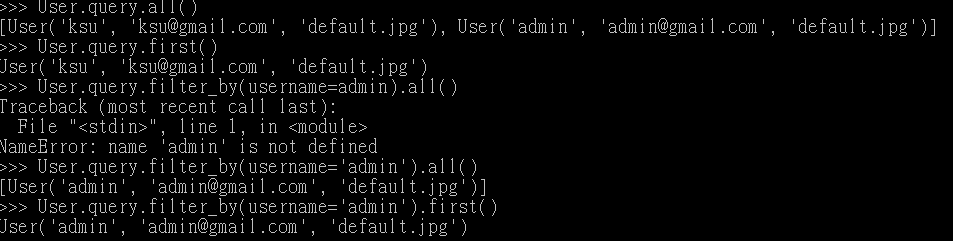
Filter:比較常使用==

Ex: user = User.query.filter(name == “Ksu”).all()

Filter\_by: 比較常使用=，但是沒有辦法使用>、<，類似SQL的where

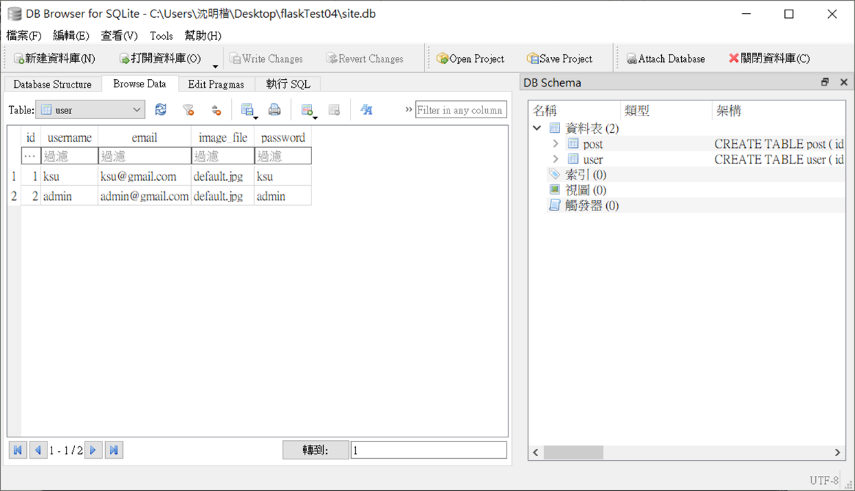
Ex: user = User.query.filter\_by(name=”Ksu”).first()

**查詢執行函數:all()、first()回傳第一筆資料，無資料回傳none、get()、first\_or\_404()、get\_or\_404()……等**

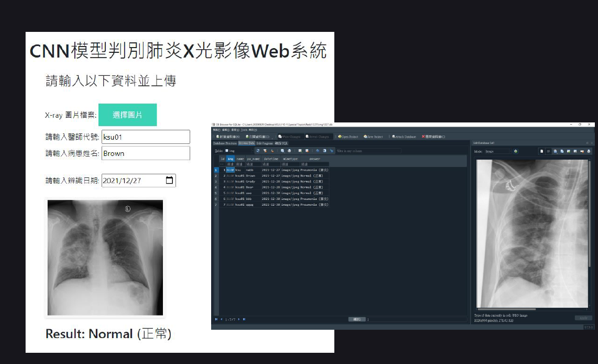
****

**SQLite管理工具**

DB Browser for SQLite:圖形化管理資料庫工具，可以設定資料庫欄位，本專題透過此軟體來管理SQLite資料庫，例如:醫師的id及密碼、醫師上傳的X光片(x-ray)以BLOB(Binary Large Object)的格式儲存在資料庫、上傳時間或是辨識結果……等。



* BLOB(Binary Large Object) 用來儲存二進位檔案，像是影音、圖片或是大型文件，本專題利用BLOB格式儲存醫師從前端所上傳的X光片影像。

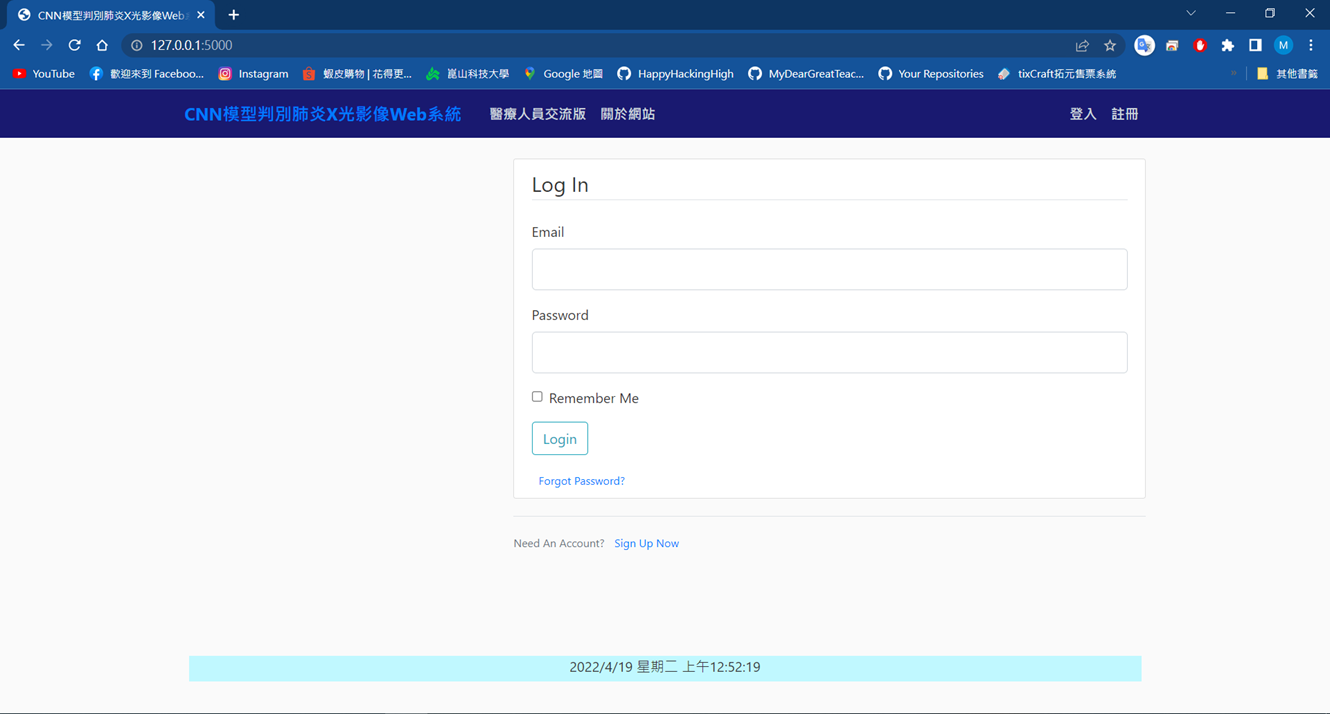


(醫師上傳x光至資料庫)

**本專題網頁伺服(Web Server)**

當使用者(醫師)透過前端介面上傳X光影像，上傳的影像會透過Web server放到以訓練好的CNN辨識訓練模型(.h5檔)執行辨識，辨識完結果會再回傳至使用者Web前端介面。讓使用者得知目前的辨識結果(正常或肺炎)，資料庫透過SQLalchemy使用SQLite來管理資料庫的部分，X光影像與其他資訊會上傳至資料庫儲存，使用者與使用者之間是隔開的，並不能使用其他使用者的身分來操作，來保護隱私。

(1) 使用者(醫師)登入介面，讓使用者有專屬的帳號密碼，提供安全性。這個網站必須要由相關醫師才能進入，而非開放式的網站，讓病患的醫療資訊曝光。



(2) 使用者(醫師)登入後的介面，紅框處為該帳密使用者名稱會自動帶入，例如:登入的使用者名稱為ksu，系統會顯示歡迎登入ksu醫師而不會顯示其他人；若登入的使用者為ksu1，則顯示歡迎登入ksu1醫師；ksu跟ksu1是不同使用者，故無法與用對方的身分去進行其他進一步的操作。

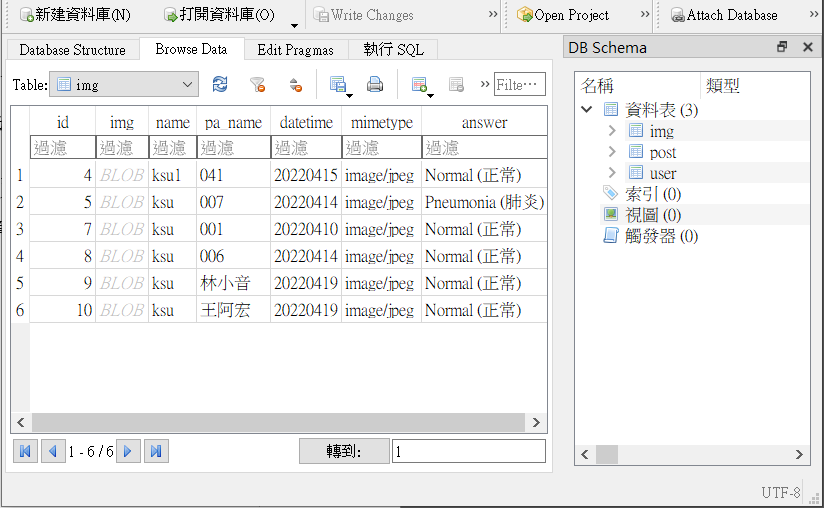




(3)使用者(醫師)點擊sidebar功能列的辨識X光片來上傳患者肺部影像X光透過Web前端介面上傳至Web server(網頁伺服器)，X光影像、病患姓名、上傳時間及辨識結果正常(NORMAL)或是肺炎( PNEUMONIA)會儲存到資料庫(Database)。X光影像是以BLOB的格式來儲存圖片。



(辨識X光片介面)



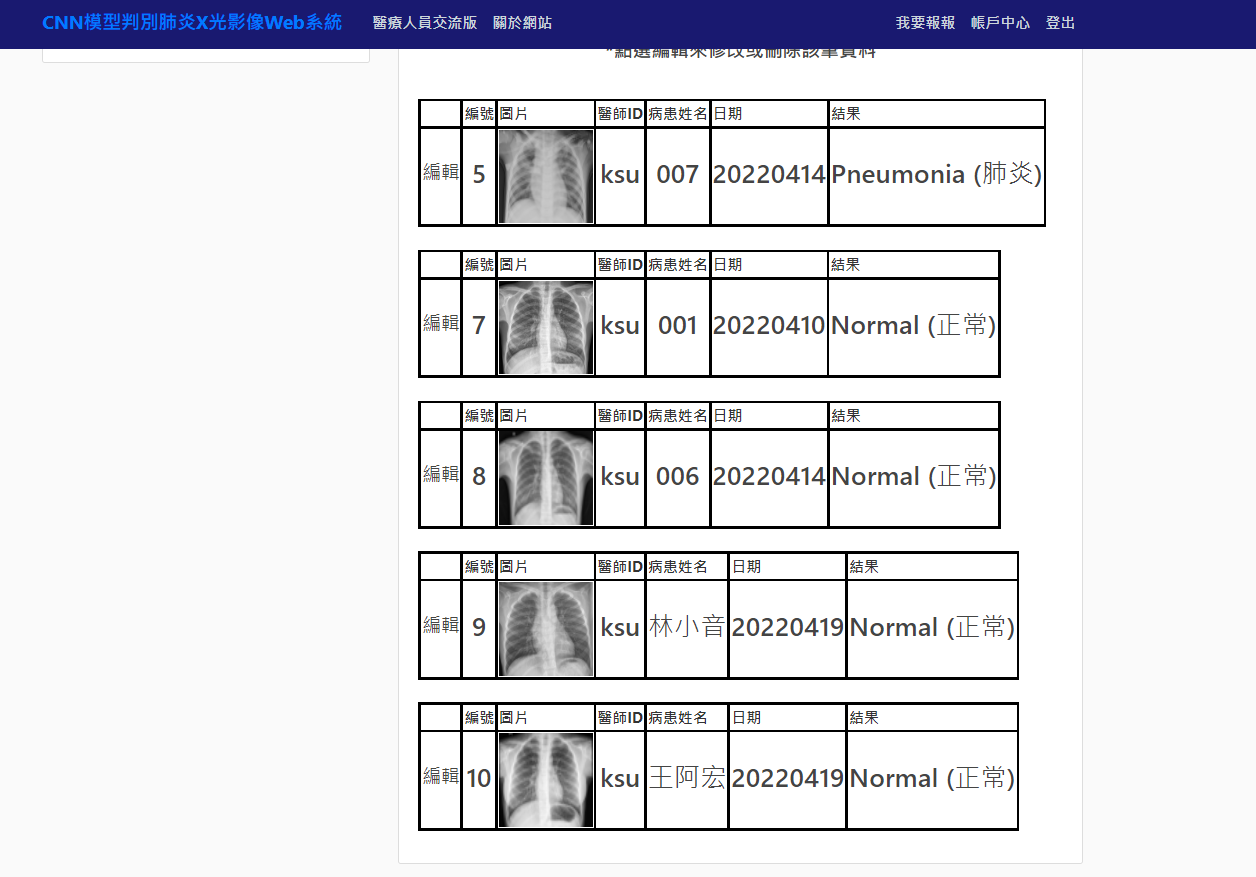
(可發現資料已上傳至資料庫db)

(4)使用者(醫師)可查詢患者肺部影像X光辨識紀錄，但本系統是無法查詢到別人的辨識紀錄，能提供安全性。本系統的輸入框會自動帶入使用者名稱，透過使用者名稱去查詢該使用者的辨識歷史紀錄，使用者無法去更改輸入框，故無法去查詢別人的歷史紀錄來進一步保護隱私。點選查詢按鈕後會列出表格(table)即是辨識歷史紀錄，表格最左邊會有編輯按鈕，下一個步驟會解釋編輯按鈕功能。

例如身分為ksu:



(辨識記錄查詢介面)



(辨識紀錄列表)

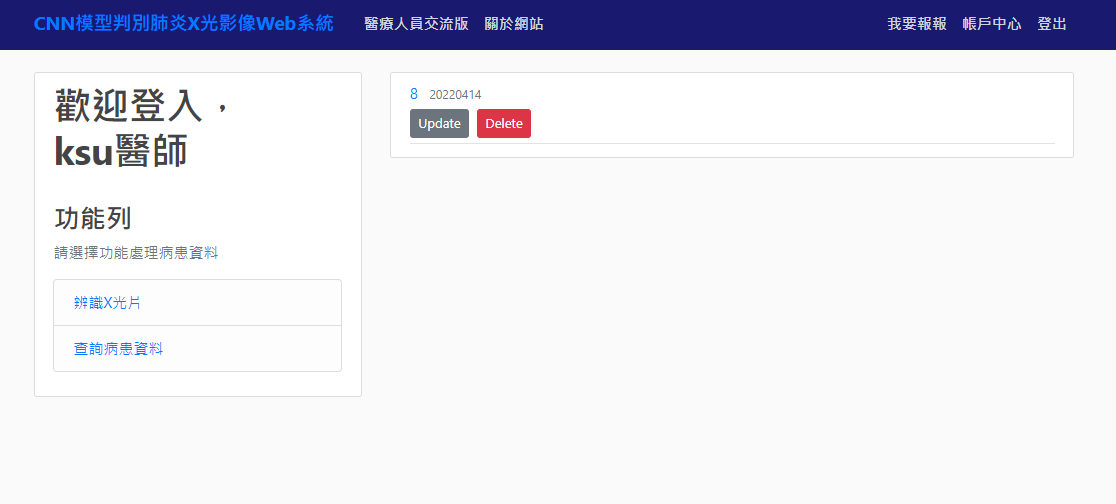
例如身分為ksu1:



(5)使用者(醫師)點擊表格最左邊的編輯會看到Update(更新)與Delete(刪除)的兩個按鈕:

Update更新按鈕:當使用者想要修改錯誤的辨識紀錄，可點選編號左邊編輯的按鈕，即可進入編輯介面，點選Update按鈕就能夠修改病患姓名或上傳影像的日期。若成功刪除會有Flash message來提醒使用者該筆資料已更新。

例如欲更改編號8(ID 8)的資訊:



(編輯頁面畫面，並點選Update按鈕，但是只有病患姓名與上傳日期能更改)

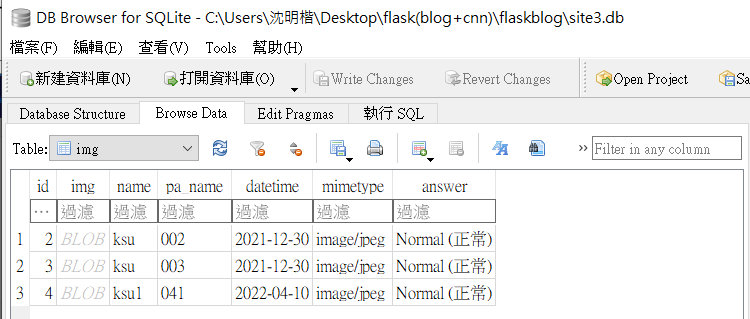


(修改病患姓名)



(Flash message顯示已更新，資料庫也更新成功)

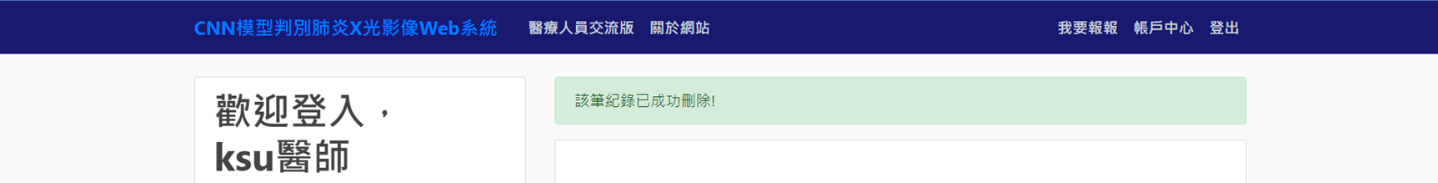
Delete刪除按鈕:使用者想要刪除某一筆辨識的歷史紀錄，可點選編號左邊編輯的按鈕，即可進入編輯介面，點選Delete刪除按鈕，系統會跳出警告進一步詢問是否要刪除該筆資料，點擊「是，我要刪除｣字樣即可刪除該筆的紀錄。資料庫也會刪除掉紀錄。



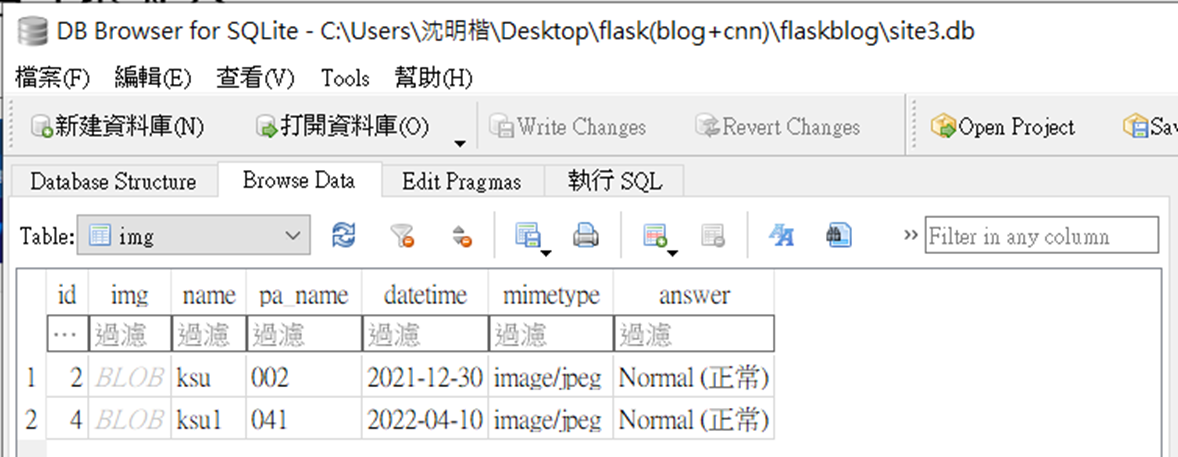
(刪除前db資料)



(系統alert警告是否刪除該筆資料)



(Flash message顯示該筆紀錄已刪除)



(紀錄已從db刪除)



(重新進入查詢也發現該筆紀錄已刪除)

* 結論:透過Python來使用flask架設網頁伺服器是一種蠻簡單、方便的一種方式，對於新手來說也容易操作與執行，Python是目前蠻火紅的一種程式語言，語法本身也是適合程式的新手，路由(Routes)、網頁模板(templates)、權限(authorization)這些架設網頁伺服器所需要的基本工具在flask都有，再加上SQLalchemy對資料庫進行操作，讓我們的CNN模型判別肺炎X光影像Web平台系統能順利建置成功。
* 在系統中無法藉由登入別的使用者來進行進一步的操作，無法利用別人的身分來進行新增、修改、刪除與查詢來達成維護安全性。