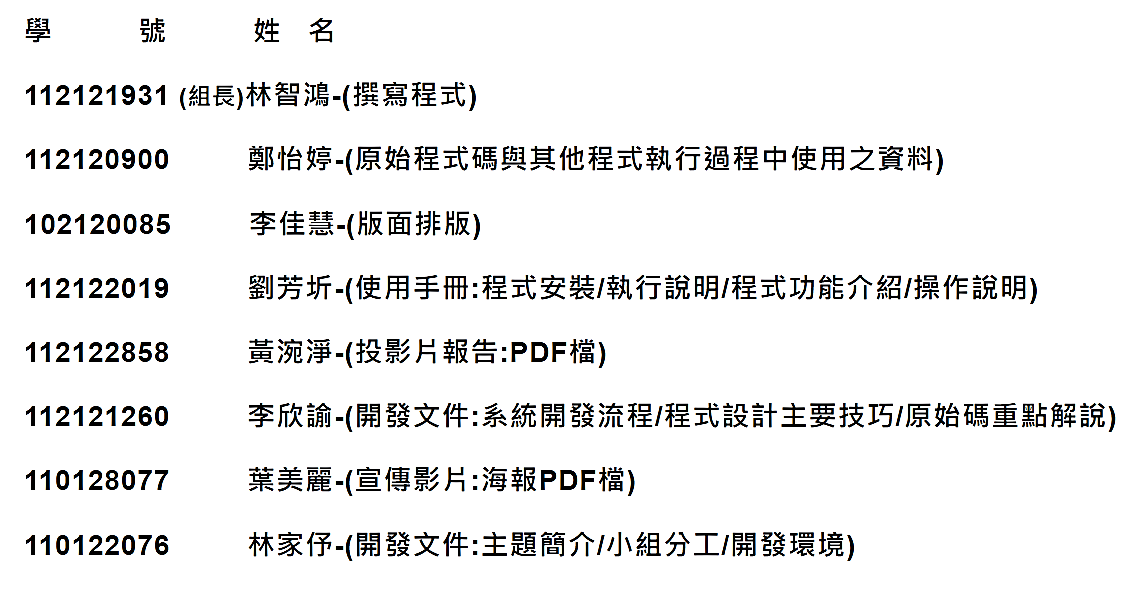
**專題報告：爬蟲系統（包含天氣查詢、房屋實價查詢、臺灣地圖顯示氣溫查詢）**

**班級：ZZZ001 小組名稱：衝吧!勇者無懼 指導老師：蕭翔之**

****

1**. 專題主題：**

**本專題為一個爬蟲系統，整合多種實用功能，包括房屋實價查詢、天氣資料查詢及臺灣各地即時氣溫顯示。系統能動態下載台灣房屋實價登錄資料，讓使用者依房價區間篩選，並結合 Google Maps 查詢位置。此外，系統也能透過 API 爬取天氣資訊，並根據查詢結果在地圖上顯示各城市的即時氣溫。**

**2. 簡介：該系統提供了以下主要功能-**

**❶ 房屋實價登錄查詢，並自動生成地理位置對應的 Google Maps 連結。**

**❷ 天氣查詢，利用爬蟲技術或 API 取得即時天氣資訊。**

**❸ 臺灣地圖顯示氣溫，依據各城市的經緯度，在地圖上顯示最新氣溫數據。**

**3. 開發工具與環境：**

**🞛開發工具🞛**

**❶ 程式語言：Python 3.12.5**

**❷ 開發框架：Flask（提供網頁應用的基礎結構）**

**❸ 資料處理：pandas 用於處理和分析實價登錄資料**

**❹ 外部套件：**

**⦿ requests：用於進行 HTTP 請求以下載資料或查詢天氣 API**

**⦿ zipfile：用於解壓縮實價登錄資料檔案🞛程式及路由設計**

**⦿ pandas：處理資料表格操作，如篩選和轉換**

**⦿ Flask：建立網頁應用程式及路由設計**

**🞛開發環境🞛**

**❶ 作業系統：Windows 10 / Linux**

**❷ 開發工具：Visual Studio Code / PyCharm**

**❸ 版本控管：Git**

**4. 系統開發流程：**

**🞛流程步驟🞛**

**❶ 需求分析：確定系統功能，包括房屋實價查詢、天氣查詢和即時氣溫顯示。**

**❷ 系統設計：設計資料爬取流程與數據處理邏輯，定義系統功能模組。**

**❸ 開發階段：**

**⦿ 建立 Flask 架構的網頁伺服器**

**⦿ 設計下載房屋實價登錄資料的功能，並實現資料篩選功能**

**⦿ 連接天氣 API，並將結果顯示在臺灣地圖上**

**❹ 資料整合：爬取房屋資料與天氣數據，整合至網頁應用中。**

**❺ 測試與調整：確保系統穩定運行，並優化顯示效果及查詢速度。**

**5. 程式設計主要技巧：**

**❶ 爬蟲技術：使用 requests 庫來爬取資料，下載實價登錄檔案並解壓縮**

**處理。**

**❷ 資料篩選與轉換：使用 pandas 篩選資料，例如篩選符合特定價格範圍的房屋交易，並將地址轉換為可點擊的 Google Maps 連結。**

**❸ 地理位置處理：系統內建台灣各縣市的經緯度數據，結合房屋地址與天氣查詢結果生成地圖顯示。**

**❹ 網頁前端顯示：利用 Flask 搭建簡易網頁應用，顯示篩選後的房屋交易與天氣查詢結果。**

**❺ API 整合：透過 HTTP 請求調用外部天氣 API，並將數據格式化後顯示在網頁中。**

**6. 原始碼重點解說：**

**❶ 房屋實價查詢系統主要邏輯：**

**🞛 下載與解壓資料 🞛**

**⦿ 此函數用於下載最新的實價登錄資料，並將其解壓縮到指定的資料夾中。**

**def download\_and\_extract\_data():**

**response = requests.get(ZIP\_URL)**

**with open(ZIP\_FILE\_PATH, "wb") as zip\_file:**

**zip\_file.write(response.content)**

**with zipfile.ZipFile(ZIP\_FILE\_PATH, "r") as zip\_ref:**

**zip\_ref.extractall(DATA\_DIR)**

**🞛 篩選資料與生成 Google Maps 連結 🞛**

**⦿ 使用者可透過城市、最低與最高房價來篩選資料，並生成房屋的 Google Maps 連結。**

|  |
| --- |
| **def query\_real\_estate(city, min\_price, max\_price):**  **df = read\_city\_data(city\_files[city])**  **filtered\_df = df[(df['總價元'] >= min\_price \* 1000000) & (df['總價元'] <= max\_price \* 1000000)]**  **df['土地位置建物門牌'] = df['土地位置建物門牌'].apply(lambda x: f'<a href="{generate\_google\_maps\_link(x)}" target="\_blank">{x}</a>')**  **return filtered\_df[['鄉鎮市區', '土地位置建物門牌', '總價元', '單價元平方公尺']].to\_html(escape=False, render\_links=True)** |

**❷ 天氣查詢邏輯：**

**⦿ 使用 requests 庫查詢天氣 API，並將結果格式化為可在地圖上顯示的格式。**

**❸ 系統運行流程：**

**⦿ 程式啟動後，Flask 會開啟網頁伺服器，使用者可透過網頁輸入查詢條件，如城市與價格區間，系統會即時回應並顯示結果。**