|  |
| --- |
| 班級：ZZZ001 小組名稱：衝吧!勇者無懼 指導老師：蕭翔之 |
| 專題報告開發文件 |
| 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、各地新聞彙整） |

|  |
| --- |
| 小婷 鄭  2024/11/3 |

目錄

[一、 小組成員 4](#_Toc182337285)

[二、 專題主題： 5](#_Toc182337286)

[三、 簡介： 5](#_Toc182337287)

[四、 開發工具與環境： 6](#_Toc182337288)

[開發工具與套件 6](#_Toc182337289)

[開發環境 7](#_Toc182337290)

[五、 系統開發流程： 8](#_Toc182337291)

[**1.** **需求分析：確定系統的主要功能和目標。** 8](#_Toc182337292)

[**2.** **可行性確認：** 8](#_Toc182337293)

[**3.** **系統設計：** 9](#_Toc182337294)

[**4.** **開發環境建立：** 9](#_Toc182337295)

[5. **功能開發：** 10](#_Toc182337296)

[**6.** **程式設計主要技巧：** 11](#_Toc182337297)

[(1) 模組化設計 11](#_Toc182337298)

[(2) 使用框架 11](#_Toc182337299)

[(3) 資料處理與清理 12](#_Toc182337300)

[(4) 視覺化 12](#_Toc182337301)

[(5) API 整合 13](#_Toc182337302)

[(6) 非同步處理 13](#_Toc182337303)

[(7) 錯誤處理 14](#_Toc182337304)

[(8) 使用模板引擎 14](#_Toc182337305)

[**7.** **原始碼重點解說：** 16](#_Toc182337306)

[(1) 主程式 (`main.py`) 16](#_Toc182337307)

[(2) HTML 表單頁面 (`index.html`) 17](#_Toc182337308)

[(3) 泡泡圖繪製 (`bubbles.py`) 17](#_Toc182337309)

[(4) 台灣縣市經緯度資料 (`cities.py`) 18](#_Toc182337310)

[(5) 貸款計算 (`loancalc.py`) 18](#_Toc182337311)

[(6) 新聞爬蟲 (`news.py`) 19](#_Toc182337312)

[(7) 實價登錄資訊下載與查詢 (`realestate.py`) 20](#_Toc182337313)

[(8) 房地產地圖查詢 (`realestatemap.py`) 20](#_Toc182337314)

[(9) 天氣資訊查詢 (`weather.py`) 21](#_Toc182337315)

# 小組成員

|  |  |  |
| --- | --- | --- |
| 學號 | 姓名 | 工作 |
| 112121931 | 林智鴻 | 架構規劃、技術支援、GitHub建置與管理、主程式、查詢指定地點實價登錄資訊地圖、查詢台灣各地天氣、查詢區域新聞、開發文件校訂 |
| 112120900 | 鄭怡婷 | 需求分析與功能確認、資料源收集、可行性評估、頁面設計、下載實價登錄資訊、查詢指定地點實價登錄資訊、列印功能、海報、開發文件撰寫 |
| 102120085 | 李佳慧 | 需求分析與功能確認、可行性評估、地點實價登錄資訊泡泡圖 |
| 112122019 | 劉芳圻 |  |
| 112122858 | 黃涴淨 | 貸款計算、海報 |
| 112121260 | 李欣諭 |  |
| 110128077 | 葉美麗 | 海報 |

# 專題主題：

本專題以台灣為核心，整合多種實用功能與資訊，包括房屋實價查詢、統計圖表、台灣各地即時氣溫顯示及地方即時新聞等。專題使用Python作為後端開發語言，建立網頁應用程式，並整合多個資訊來源。透過各種套件的運用，如Bootstrap、Folium、Pandas等，截取並呈現資料，並使用GitHub進行程式碼管理。此專題不僅讓小組成員深入了解Python，還能增進他們對應用程式開發的認識。

# 簡介：

1. **房屋實價登錄查詢：**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。系統會分析實價登錄資料，並以各式圖表的方式展示，提供使用者更直觀的數據視覺化。
2. **實價登錄資訊地圖：**以地圖方式呈現實價登錄資訊。
3. **查詢台灣各地天氣：**透過 API 與地圖系統整合，顯示各縣市即時天氣資訊。使用者可以輕鬆查詢當前天氣狀況，並獲得詳細的氣象數據。
4. **各地即時新聞：**系統會截取各地即時新聞，無廣告且提升瀏覽速度。使用者可以快速瀏覽新聞摘要，對有興趣的內容，仍可透過聯結檢視原新聞內容。
5. **貸款試算：**提供房屋貸款試算功能，使用者可以輸入貸款金額、利率及還款期限，系統會自動計算每月還款金額及總利息，幫助使用者更好地規劃財務。
6. **列印：**所有查詢結果能進行列印，或輸出成PDF格式儲存。

# 開發工具與環境：

## 開發工具與套件

|  |  |
| --- | --- |
| 工具與版本 | 說明 |
| Python 3.12.5 | 使用程式語言與版本  <https://www.python.org/downloads/> |
| Visual Studio Code 1.93.1 | 主要開發IDE工具，與版本簽入GitHub。  [Download Visual Studio Code Mac, Linux, Windows](https://code.visualstudio.com/Download) |
| GitHub | 版本控制軟體，並於每次Commit時自動透過Pylint進行檢核。  [GitHub 存放路徑](https://github.com/112121931/PY/tree/main/Source/midtermwork) (未開放) |
| GitHub DeskTop | **GitHub Desktop** 是一個免費且開源的應用程式，旨在簡化使用 Git 進行版本控制的流程。  [Download GitHub Desktop | GitHub Desktop](https://desktop.github.com/download/) |
| Pylint 3.3.1 | 確保程式開發風格符合PEP8標準。  無使用其它套件。 |
| Flask 3.0.3 | Flask 是一個輕量級的 Python 網頁框架，適合快速開發網頁應用程式。 |
| Folium 0.17.0 | Folium 是一個用於地理數據可視化的 Python 套件，基於 Leaflet.js。 |
| Ipywidgets 8.1.5 | Ipywidgets 是 Jupyter Notebook 的互動小工具庫，用於創建交互式小部件。 |
| Ipython 8.27.0 | Ipython 是一個強大的互動式 Python Shell，提供了豐富的工具和功能。 |
| Pandas 2.2.2 | Pandas 是一個強大的數據處理和分析工具，提供了高效的數據結構和操作方法 |
| bs4 0.0.2 | bs4 (Beautiful Soup) 是一個用於解析 HTML 和 XML 文件的 Python 庫。 |

## 開發環境

作業系統：Windows 10 / 11

# 系統開發流程：

1. **需求分析：確定系統的主要功能和目標。**

經過小組討論，我們決定開發一個爬蟲系統，旨在收集並呈現多種資料。具體功能包括：

* 房屋實價查詢：動態下載並分析台灣房屋實價登錄資料，提供使用者依房價區間篩選和地圖查詢功能。
* 天氣查詢：整合即時天氣資料，顯示各縣市的天氣狀況，並提供詳細的氣象數據。
* 即時氣溫顯示：顯示台灣各地的即時氣溫，讓使用者能夠快速了解當前氣溫狀況。

1. **可行性確認：**

* **資料來源評估：**尋找並評估相關的資料來源，確保資料的可用性和準確性。主要資料來源包括：
* [內政部不動產成交案件實際資訊資料供應系統](https://plvr.land.moi.gov.tw/Index)：提供房屋實價登錄資料。
* [交通部中央氣象](https://www.cwa.gov.tw/V8/C/)：提供即時天氣資訊。
* [即時影像攝影機](https://tw.live/)：提供各地即時影像資料。
* [自由時報電子報](https://news.ltn.com.tw/)：提供即時新聞資料。
* [Google 地圖](https://www.google.com.tw/maps)：提供地理位置和地圖服務。
* **技術可行性驗證：**進行初步的程式撰寫和測試，驗證技術方案的可行性。確保各資料來源能夠順利整合到系統中，並能夠穩定地提供所需的資料。

1. **系統設計：**

* **框架選擇：**使用 Flask 框架開發網頁應用程式，確保系統的靈活性和可擴展性。
* **前端設計：**前端使用 HTML 和 Bootstrap 完成響應式頁面設計，主要負責用戶輸入和顯示查詢結果。確保頁面在各種設備上都能良好顯示。
* **後端開發：**後端使用 Python 進行開發，負責資料的查詢和計算。將結果以表格、圖片和地圖的方式組成頁面，並回傳給前端顯示。
* **開發架構：**遵循 MVC（ModelViewController）開發架構，以保持程式碼的簡潔和易於維護。這樣的架構有助於分離資料處理、業務邏輯和顯示層，提升系統的可維護性。
* **使用者體驗：**系統設計以簡單易用為前提，確保使用者能夠輕鬆上手並高效使用各項功能。

1. **開發環境建立：**

* **版本控制：**使用 GitHub 建立版本控制系統，以便進行小組內程式碼、文件等管理作業。透過 GitHub 整合 Pylint，確保程式開發風格符合標準，提升程式碼品質。
* **開發工具：**選擇 Visual Studio Code 作為主要開發工具，提供豐富的擴展功能和便捷的開發環境。
* **系統安裝文件：**建立詳細的系統安裝文件，供小組成員參考，確保每個成員都能順利設置開發環境。這些文件也將用於後續的系統維護和升級，提供清晰的指導。

1. **功能開發：**

* **分階段開發：**根據需求分析和系統設計，分階段開發各項功能模組，包括：
* **房屋實價登錄查詢：**動態下載並分析台灣房屋實價登錄資料，提供使用者依房價區間篩選和地圖查詢功能。
* **天氣查詢：**整合即時天氣資料，顯示各縣市的天氣狀況，並提供詳細的氣象數據。
* **即時新聞顯示：**截取各地即時新聞，無廣告且提升瀏覽速度，使用者可以快速瀏覽新聞摘要，並透過聯結檢視原新聞內容。
* **貸款試算：**提供房屋貸款試算功能，使用者可以輸入貸款金額、利率及還款期限，系統會自動計算每月還款金額及總利息，幫助使用者更好地規劃財務。
* **單元測試：**每個功能模組開發完成後，進行單元測試，確保功能正常運作。
* **程式碼品質檢查：**使用 Pylint 工具進行程式碼品質檢查，確保程式碼符合標準風格和最佳實踐。Pylint 會自動檢查程式碼中的錯誤、潛在問題和不符合規範的部分，幫助開發者維持高品質的程式碼。

1. **程式設計主要技巧：**

### 模組化設計

#### **說明：**

將程式分割成獨立的模組，每個模組負責特定的功能。這樣可以提高程式的可維護性和可擴展性。

例如，在你的專案中，將房地產查詢、泡泡圖繪製、貸款計算等功能分別放在不同的檔案中。

#### **優點：**

易於維護：每個模組可以單獨測試和修改，不會影響其他部分。

可重用性：模組化的程式碼可以在其他專案中重用。

### 使用框架

#### **說明：**

使用像 Flask 這樣的框架來建構網頁應用程式，可以簡化路由和請求處理的實現。

Flask 提供了簡單而強大的工具來處理 HTTP 請求、模板渲染和靜態文件服務。

#### **優點：**

快速開發：框架提供了許多內建功能，減少了開發時間。

社區支持：流行的框架通常有大量的文檔和社區支持，遇到問題時可以快速找到解決方案。

### 資料處理與清理

#### **說明：**

在處理數據時，進行資料清理和格式轉換，確保數據的準確性和一致性。

例如，在處理房地產數據時，移除缺失或無效的數據，並將數字格式化為帶千分位的格式。

#### **優點：**

提高數據質量：清理數據可以減少錯誤和異常，提高數據的可靠性。

便於分析：格式化數據使其更易於分析和視覺化。

### 視覺化

#### **說明：**

使用 Matplotlib 和 Folium 等工具進行數據視覺化，幫助用戶更直觀地理解數據。

例如，繪製房地產交易數據的泡泡圖，或在地圖上顯示房地產交易位置。

#### **優點：**

直觀展示：視覺化可以將複雜的數據轉化為易於理解的圖表和地圖。

發現趨勢：視覺化有助於發現數據中的趨勢和模式。

### API 整合

#### **說明：**

通過 API 獲取外部數據（如天氣資訊），並將其整合到應用程式中，提供更豐富的功能。

例如，從氣象資訊開放平臺取得台灣各城市的天氣資訊。

#### **優點：**

擴展功能：API 使應用程式能夠訪問和使用外部服務，擴展其功能。

實時數據：API 可以提供實時數據，保持應用程式的數據最新。

### 非同步處理

#### **說明：**

使用 JavaScript 和 Fetch API 進行異步請求，提升用戶體驗，避免頁面重新載入。

例如，在表單提交後，使用 Fetch API 發送 POST 請求，並將結果顯示在 iframe 中。

#### **優點：**

提升性能：異步處理可以減少頁面重新載入的次數，提高應用程式的響應速度。

改善用戶體驗：異步請求使應用程式更加流暢和互動。

### 錯誤處理

#### **說明：**

在進行網路請求和資料處理時，加入錯誤處理機制，確保應用程式的穩定性。

例如，處理 API 請求失敗的情況，並向用戶顯示友好的錯誤訊息。

#### **優點：**

提高穩定性：錯誤處理可以防止應用程式崩潰，保持其穩定運行。

提升用戶體驗：友好的錯誤訊息可以幫助用戶理解問題並採取相應的行動。

### 使用模板引擎

#### **說明：**

使用 Jinja2 模板引擎動態生成 HTML，根據後端數據動態更新前端頁面。

例如，根據新聞爬蟲獲取的數據動態生成新聞列表頁面。

#### **優點：**

動態內容：模板引擎使應用程式能夠根據數據動態生成內容，提供更豐富的用戶體驗。

簡化開發：模板引擎提供了簡單的語法來插入數據和控制頁面結構，減少了手動編寫 HTML 的工作量。

1. **原始碼重點解說：**

### 主程式 (`main.py`)

#### **功能：**

使用 Flask 框架建構網頁應用程式，整合多種功能，包括下載實價登錄資訊、查詢房地產資訊、繪製泡泡圖、計算貸款、查詢天氣和新聞等。

#### **主要部分：**

* 設置工作目錄：確保當前工作目錄正確。
* 建立 Flask 應用程式：初始化 Flask 應用程式。
* 主頁面路由 (`/`)：顯示主頁面。
* 查詢路由 (`/query`)：處理 POST 請求，根據選項執行不同的功能：

下載實價登錄資訊 (`download\_real\_estate`)

查詢天氣 (`weather`)

查詢房地產資訊 (`real\_estate`)

查詢房地產地圖 (`real\_estate\_map`)

查詢新聞 (`news`)

繪製泡泡圖 (`bubbles`)

貸款計算 (`loancalc`)

### HTML 表單頁面 (`index.html`)

#### **功能：**

提供用戶介面，讓用戶選擇查詢類型並輸入相關參數。

#### **主要部分：**

* 使用 Bootstrap 設計響應式網頁：提供美觀且響應式的用戶介面。
* 表單選項：用戶可以選擇不同的查詢類型，如下載實價登錄資訊、查詢房地產資訊、查詢天氣、查詢新聞、繪製泡泡圖和貸款計算。
* 動態顯示輸入欄位：根據用戶選擇的查詢類型，動態顯示或隱藏相關的輸入欄位（如地點、價格範圍、貸款參數等）。
* JavaScript 功能：處理表單提交和結果顯示，提升用戶體驗。

### 泡泡圖繪製 (`bubbles.py`)

#### **功能：**

使用 Matplotlib 繪製房地產交易數據的泡泡圖。

#### **主要部分：**

* 資料預處理：根據城市和價格範圍篩選數據，清理數據，移除缺失或無效的數據。
* 計算單價元平方公尺：計算每個房屋的單價，並按區域分組計算交易總數。
* 繪製泡泡圖：使用 Matplotlib 繪製泡泡圖，設定圖表標題、軸標籤和圖例，並將圖表保存為 base64 編碼的字符串。

### 台灣縣市經緯度資料 (`cities.py`)

#### **功能：**

提供台灣各縣市的經緯度資料，並包含輔助函數來取得城市清單和對應的資料檔案名稱。

#### **主要部分：**

* 經緯度資料：定義 `cities\_data` 字典，包含各縣市的經緯度。
* 輔助函數：
* `get\_cities()`：返回包含經緯度資料的字典。
* `get\_location\_by\_city(city\_name)`：根據城市名稱返回對應的經緯度。
* `get\_city\_files()`：返回城市名稱與對應資料檔案名稱的字典。

### 貸款計算 (`loancalc.py`)

#### **功能：**

計算貸款的每月還款金額，並生成詳細的還款計劃表。

#### **主要部分：**

* 計算每月還款金額：根據貸款金額、年利率和貸款期限計算每月還款金額。
* 生成還款計劃表：生成包含期數、未償還貸款、償還本金、支付利息和每月供款的還款計劃表，並將其轉換為 HTML 表格。

### 新聞爬蟲 (`news.py`)

#### **功能：**

爬取自由時報區域新聞，根據用戶選擇的地點顯示對應的新聞標題和內容。

#### **主要部分：**

* 定義地點選項：定義 `locations` 字典，包含各地點的新聞頁面 URL。
* 取得新聞內容：使用 BeautifulSoup 解析 HTML，提取新聞段落內容，過濾掉不相關的段落。
* 查詢新聞清單：根據地點選項獲取對應的新聞頁面 URL，提取新聞標題和鏈接，並渲染到 `News.html` 模板中。

### 實價登錄資訊下載與查詢 (`realestate.py`)

#### **功能：**

下載並解壓縮實價登錄資料，根據城市和價格範圍查詢房地產交易資料，並生成 Google Maps 連結。

#### **主要部分：**

* 下載並解壓縮資料：定義 `download\_and\_extract\_data()` 函數下載並解壓縮資料。
* 讀取指定城市的 CSV 資料：定義 `read\_city\_data()` 函數讀取指定城市的 CSV 資料。
* 查詢房地產交易資料：定義 `query\_real\_estate()` 函數根據城市和價格範圍查詢房地產交易資料，並生成 HTML 表格顯示結果。

### 房地產地圖查詢 (`realestatemap.py`)

#### **功能：**

使用 Folium 繪製房地產交易資料地圖，根據城市和價格範圍篩選數據並在地圖上添加標記。

#### **主要部分：**

* 清理地址：定義 `clean\_address()` 函數清理地址，移除不必要的部分。
* 查詢座標資訊：定義 `get\_coordinates()` 函數使用 Geopy 的 Nominatim 服務查詢地址的經緯度，並使用暫存檔來儲存和查詢已經獲取的座標資訊。
* 查詢房地產交易資料地圖：定義 `query\_real\_estate\_map()` 函數根據城市和價格範圍篩選房地產交易資料，並使用 Folium 繪製地圖，添加標記顯示每筆交易的門牌和金額。

### 天氣資訊查詢 (`weather.py`)

#### **功能：**

從氣象資訊開放平臺取得台灣各城市的天氣資訊，並在地圖上顯示。

#### **主要部分：**

* 查詢天氣資訊：定義 `get\_weather()` 函數發送 HTTP 請求到氣象資訊開放平臺 API，根據城市名稱查詢天氣資訊，並解析回應提取相關天氣資訊。
* 繪製天氣地圖：定義 `get\_weather\_map()` 函數使用 Folium 創建地圖，對每個城市添加標記，標記上顯示該城市的天氣資訊，並將地圖轉換為 HTML 格式返回。