< Dual-Source Web Scraping >

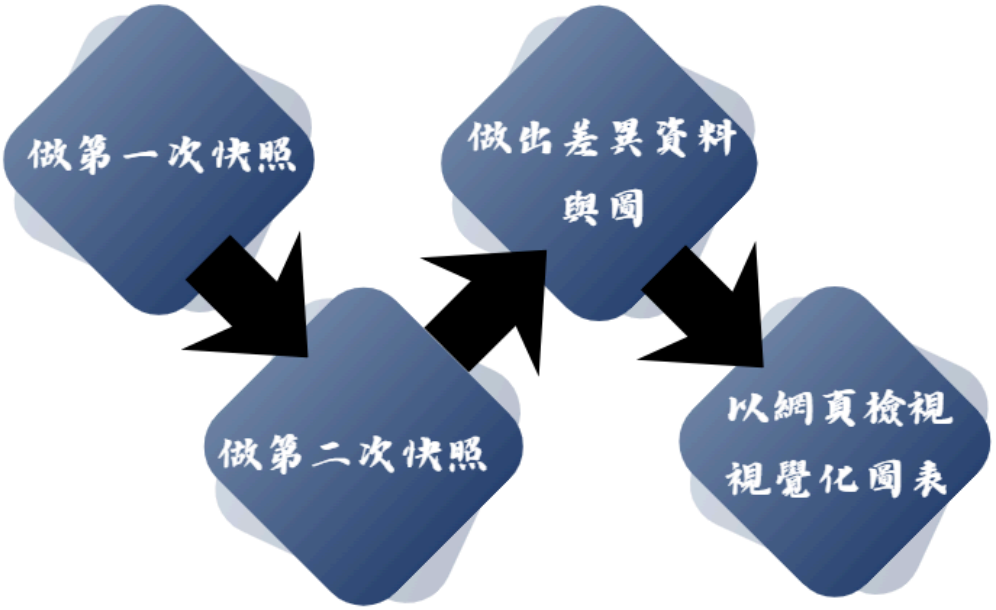
第四組

成員:
A1115505 鄒邱昂
A1115533 郭俊逸
A1115543 何翰俊
A1115545 陳至炫

Project Objective（專案目標）

本專案旨在從兩個不同性質的網站來源（1 個為靜態 HTML，1 個為需 JavaScript 渲染的動態站點）進行資料抓取，並將其儲存為結構化格式（CSV 或 SQLite）。系統需支援 incremental updates，能在第二次執行時僅抓取新增或變更項目，並輸出差異摘要（new/deleted/changed 數量）與對應的圖表。最終需提供最小化介面（CLI 或 Streamlit 簡易 UI）讓使用者可以搜尋與瀏覽資料。

System Architecture（系統架構）



(圖一)流程圖

流程圖：Data Source A/B → Scraper Engine → Data Cleaning → Storage → Diff Engine → Interface/UI

我們整個系統以模組化方式設計，從資料擷取（scrape）、清理（clean）、比對（diff）

各自封裝成獨立檔案，統一由 CLI 指令呼叫。

這樣的設計讓 pipeline 可擴充、可重複使用，同時也符合 reproducibility 的要求。

我們的系統分成三個主要模組：

A 是爬蟲與資料來源模組（scrape），

B 是清理與比對模組（clean、diff），

C 是前端介面（Streamlit）負責視覺化展示。

所有模組都以相對路徑組合，例如 data/snapshots、data/diffs、data/charts，

因此整個專案可以直接搬移，不受環境影響。

Source Selection（資料來源）

📂 pipeline/
├── scrape.py ← 抓資料
├── clean.py ← 整理格式
├── diff.py ← 比對快照
├── storage.py ← 輸出管理

Source A：靜態 HTML 網站 — 可直接使用 requests 搭配 BeautifulSoup 擷取資料。
Source B：動態渲染網站 — 需透過 Selenium 或 Playwright 模擬載入並抓取元素。
每個來源需至少蒐集 100 筆資料，並紀錄 source、id、title、url、price/date 等欄位。

我們日前在簡報上示範的兩個網站都是靜態來源，但整個架構已支援動態爬蟲(LINE TODAY)，只要在 config 裡修改設定即可。

系統會自動判斷來源型態：

如果是靜態網站，就用 requests 搭配 BeautifulSoup 抓取 HTML；

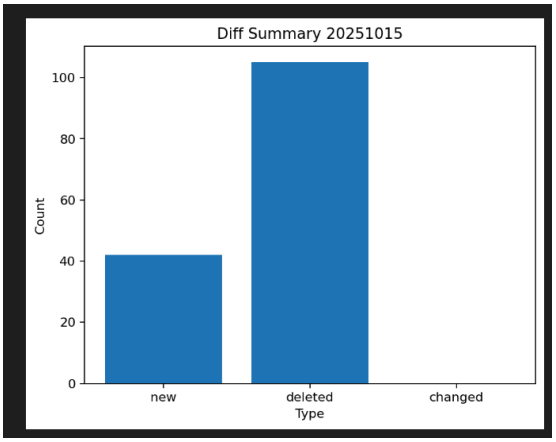
如果是動態網站，則會呼叫我們預留的 dynamic 模式（可接 Playwright 或 Selenium）。

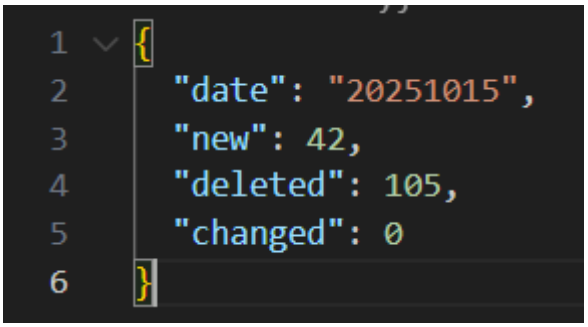
所以不論是要抓新聞、商品還是開放資料，只要修改設定檔就能擴充新的來源，這樣的設計讓整個系統彈性高、擴充性強。

補充: LINE TODAY已實現(一開始想做YOUTUBE 但被擋)
可於GITHUB分支查看此程式

diff_20251015_new.csv

```
pk,id,title,url,author,category,date,price,source,last_seen_at
linetoday_deep::https://today.line.me/tw/v3/article/0MmqpY2,https://today.line.me/tw/v3/article/0MmqpY2,02快訊／來這招？突擊成真！新
linetoday_deep::https://today.line.me/tw/v3/article/2Dj86ya,https://today.line.me/tw/v3/article/2Dj86ya,18自己先Thank You自己！獅隊公
linetoday_deep::https://today.line.me/tw/v3/article/2Dj8gPP,https://today.line.me/tw/v3/article/2Dj8gPP,164星座年底前要發了！橫財不斷
linetoday_deep::https://today.line.me/tw/v3/article/2Dj8p5a,https://today.line.me/tw/v3/article/2Dj8p5a,,https://today.line.me/tw/v
linetoday_deep::https://today.line.me/tw/v3/article/2Dj8wwa,https://today.line.me/tw/v3/article/2Dj8wwa,1737歲女星驚喜宣布「結婚、懷孕
linetoday_deep::https://today.line.me/tw/v3/article/3NwXyNy,https://today.line.me/tw/v3/article/3NwXyNy,小兒子犯罪被捕2次！向太自責「
linetoday_deep::https://today.line.me/tw/v3/article/5yN9JQR,https://today.line.me/tw/v3/article/5yN9JQR,台女赴德被當成中國通緝犯 小房
linetoday_deep::https://today.line.me/tw/v3/article/5yNOaEy,https://today.line.me/tw/v3/article/5yNOaEy,台灣男足亞洲盃慘輸泰國！總教練
linetoday_deep::https://today.line.me/tw/v3/article/5yNOxEq,https://today.line.me/tw/v3/article/5yNOxEq,國中生「帶麥當勞進星巴克」爽吃
linetoday_deep::https://today.line.me/tw/v3/article/7NzLPKl,https://today.line.me/tw/v3/article/7NzLPKl,張柏芝復出拍廣告 網驚：臉不一
linetoday_deep::https://today.line.me/tw/v3/article/8nyvE1R,https://today.line.me/tw/v3/article/8nyvE1R,,https://today.line.me/tw/v
linetoday_deep::https://today.line.me/tw/v3/article/9m731mE,https://today.line.me/tw/v3/article/9m731mE,范世錡無預警承認「我殺了于朦朧
linetoday_deep::https://today.line.me/tw/v3/article/9m7wMrx,https://today.line.me/tw/v3/article/9m7wMrx,10月15日東海Oppa生日快樂！20
linetoday_deep::https://today.line.me/tw/v3/article/DRkOpGW,https://today.line.me/tw/v3/article/DRkOpGW,092025年全球退休金指數排名出
linetoday_deep::https://today.line.me/tw/v3/article/GgzPJXy,https://today.line.me/tw/v3/article/GgzPJXy,13接到「0979」開頭電話！ 劉
linetoday_deep::https://today.line.me/tw/v3/article/Ggzj33P,https://today.line.me/tw/v3/article/Ggzj33P,台灣大賽前傳悲痛消息 詹子賢發
linetoday_deep::https://today.line.me/tw/v3/article/GgzjZey,https://today.line.me/tw/v3/article/GgzjZey,,https://today.line.me/tw/v
linetoday_deep::https://today.line.me/tw/v3/article/Kwmn00r,https://today.line.me/tw/v3/article/Kwmn00r,葉珂自爆隱形眼鏡連戴8天 「睡覺
linetoday_deep::https://today.line.me/tw/v3/article/Kwmv7Vk,https://today.line.me/tw/v3/article/Kwmv7Vk,19不用斷食、不用運動！日本明星
linetoday_deep::https://today.line.me/tw/v3/article/Kwmvagk,https://today.line.me/tw/v3/article/Kwmvagk,08震驚！大咖男星「不敵胰臟癌病
linetoday_deep::https://today.line.me/tw/v3/article/LXrVEan,https://today.line.me/tw/v3/article/LXrVEan,影音,https://today.line.me/
```





Data Fields & Cleaning（欄位與資料清理）

欄位名稱	說明	範例
id	唯一識別碼	A1234
source	資料來源網站	site_A / site_B
title	項目名稱	"Python Web Scraping 101"
url	原始連結	https://example.com/item1
author/vendor	作者或販售者	O'Reilly
category	類別	Books
date	日期 (YYYYMMDD)	20251008
price/value	價格或數值	420
last_seen_at	最後更新時間	2025-10-08 22:31

(圖二)清理流程圖

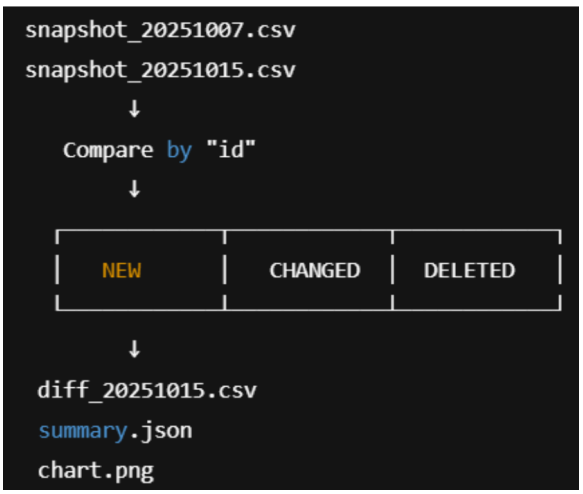
資料清理流程包含：
- Deduplication：以 id 或 url 作為主鍵，去除重複項目。
- Date Normalization：將日期轉換為統一格式（YYYYMMDD）。
- Numeric Validation：價格欄位轉為數值型態（去除$符號與千分位逗號）。
- Error Log：解析失敗的項目寫入 error_log.txt，供後續檢查。

所有欄位會統一格式並去重，確保每筆資料都能追蹤來源與更新時間。

每次抓下來的資料都會先轉成 pandas 的 DataFrame 格式，再進行欄位標準化與清理

清理後的資料更容易進行篩選、繪圖或比對，同時也避免欄位格式不一致造成的錯誤。這一步是整個系統維持資料品質的關鍵。

Incremental Update Logic（增量更新機制）



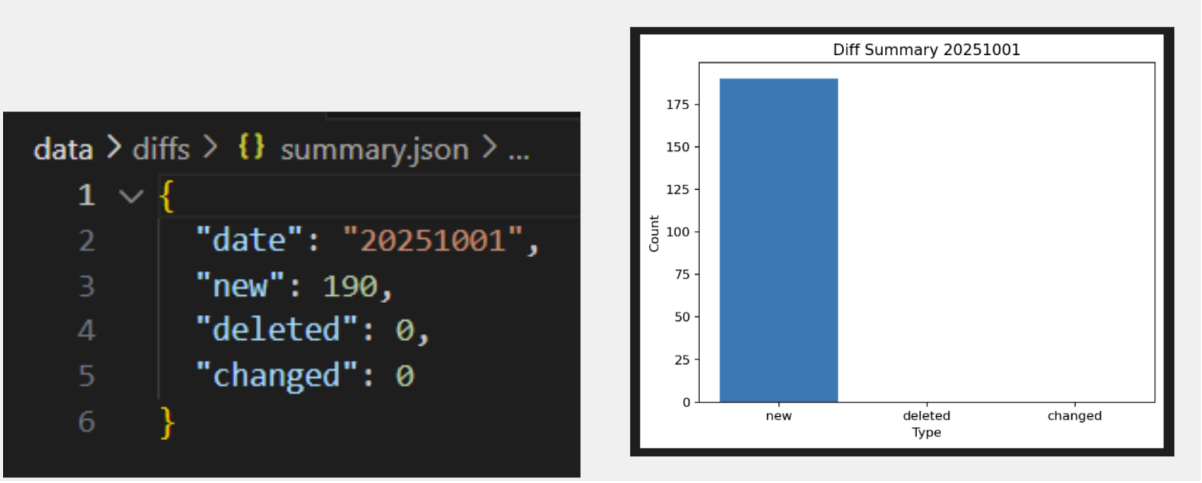
(圖三)資料處裡流程圖

系統將新快照與前一次快照進行比對，根據 id 與欄位變動內容，分為三種類型：

- 新增項目（New）
- 欄位變更（Changed）
- 已不存在於來源（Deleted）

- 比對 id + hash
比對結果會輸出為 diff_YYYYMMDD.csv 與 summary.json 供介面讀取。

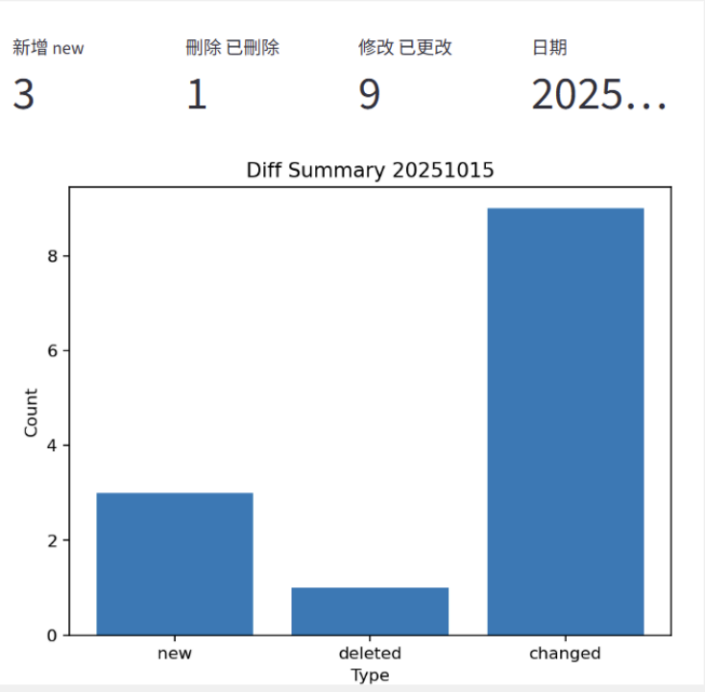
Difference Summary & Visualization（差異摘要與視覺化）



(圖四)差異摘要與圖表示意圖

系統會統計 new、deleted、changed 的數量，並使用 Matplotlib 產生長條圖（summary_*.png

）。在每次增量更新後，系統會自動統計新增、修改與刪除的項目，並產生 summary.json 及視覺化圖表。例如，這裡顯示本次更新新增 190筆



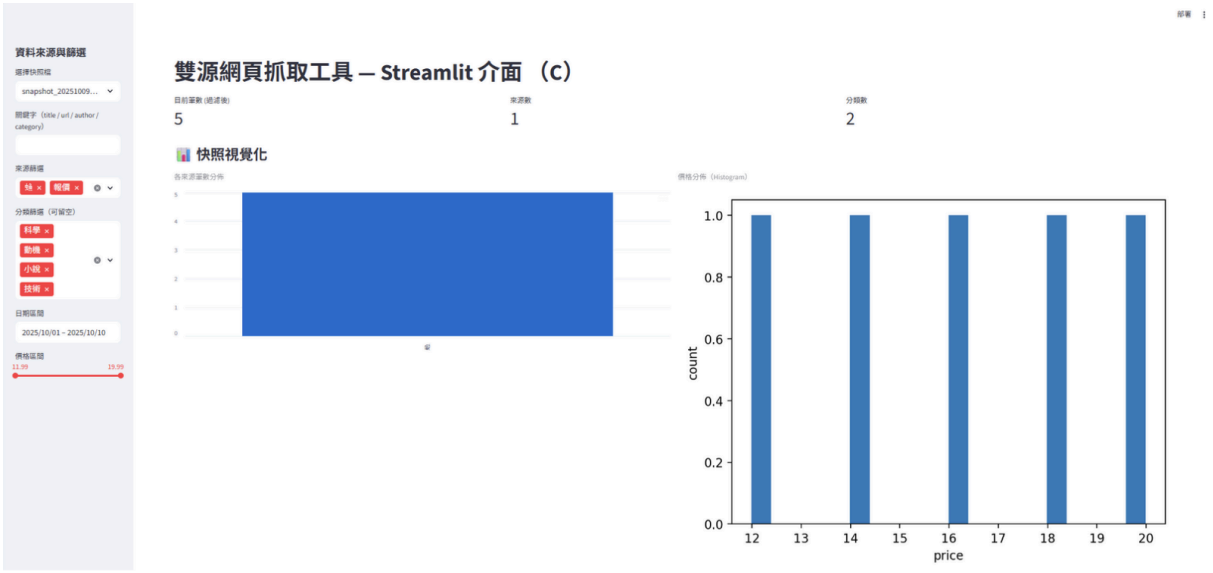
(圖五)差異圖

Diff Summary（快照：2025-10-15）：new=3、deleted=1、changed=9

比對方式：snapshot(t) ↔ snapshot(t-1)，changed＝同一 pk 至少一欄位變更

目的：快速掌握當日增量規模，驗證更新邏輯並估估審查工時

Interface Design（介面設計）



(圖六) CLI 與 Streamlit UI 示意

CLI 功能：
- python run_first_time.py（首次快照）
- python run_incremental.py（更新快照並比對差異）
- python search.py --keyword "..."（資料搜尋）
Streamlit 功能：顯示篩選控制元件、差異圖表、表格與下載按鈕。

```
# scrape
ap_scrape = sub.add_parser("scrape", help="Scrape all configured sources into a new snapshot CSV")
ap_scrape.add_argument("--config", required=True)
ap_scrape.add_argument("--out", default="data/snapshots")
ap_scrape.set_defaults(func=scrape_cmd)

# clean
ap_clean = sub.add_parser("clean", help="Clean latest snapshot (normalize date/price, dedup, last_seen_at)")
ap_clean.add_argument("--snapshots", default="data/snapshots")
ap_clean.set_defaults(func=clean_cmd)

# diff
ap_diff = sub.add_parser("diff", help="Diff latest two snapshots, write CSVs + summary.json + chart")
ap_diff.add_argument("--snapshots", default="data/snapshots")
ap_diff.add_argument("--diffs", default="data/diffs")
ap_diff.add_argument("--charts", default="data/charts")
ap_diff.set_defaults(func=diff_cmd)
```

(圖七)Command-Line Interface Design(CLI) 示意

我們設計了三個主要 CLI 指令：

scrape 負責抓取來源資料

clean 進行資料清理與格式統一

diff 則自動比較前後快照，輸出差異報表與圖表

這樣設計讓整個 pipeline 一行指令就能執行，也方便後續排程與自動化測試

Error Handling & Retry（錯誤處理與重試機制）

流程圖:try -> fail -> wait -> retry -> log
首先，我們使用指數回退的 retry 機制確保穩定性，並尊重網站的 robots.txt 限制 。系統會對 HTTP 429 / 5xx 狀態碼進行重試，採用 Exponential Backoff 機制（1s → 2s → 4s → 8s）。所有錯誤紀錄於 error_log.txt。

Reproducibility & Testing（可重現性與測試）

專案附有 run_first_time.sh 與 run_incremental.sh，確保流程可重現。
測試部分採用 pytest ，有以下四種:

test_selectors.py:驗證設定檔載入與格式正確性

test_dedup.py:測試去重功能

test_validation.py：測試資料清理與格式化函式

test_diff.py：測試快照比對功能

Results & Evaluation（成果與評估）

成功爬取兩站超過 200 筆資料，並於第二次更新正確偵測到資料差異。

在結果呈現階段，我們使用 Streamlit 實作了即時更新的 KPI 與資料表。

系統會根據篩選條件重新計算筆數、來源數與分類數，並顯示在上方指標區。

同時提供互動式表格與下載按鈕，讓使用者能快速檢視或匯出目前篩選後的資料。

所有元件都會在資料狀態改變時即時重新更新。

KPI：直接對 filtered 計數（筆數、來源數、分類數）

資料表：顯示常用欄位；提供下載過濾後 CSV（就是 filtered[...] 的當前視圖）

即時變化的原因：任何控件狀態改變 → Streamlit 重新執行腳本 → filtered 重算 → KPI/表/圖同步更新。

Limitations & Future Work（限制與未來展望）

目前限制：
- 動態網站渲染耗時，易觸發反爬機制

- 須手動設定 selectors，維護負擔高

未來改進方向：
- 引入 headless browser 優化渲染效率
- 加入 LLM-assisted selector 推薦與自動化設定
- 建構通知系統（Email / Discord webhook）