

# Markdown Browser 報告

## 摘要

有感於 md 檔案多在 hackmd、github 等網站才能完整的看見效果，為了閱讀方便，開發了 Markdown Browser。

Markdown Browser 是一款專用於 md 檔案的文件瀏覽器。可以在離線的狀況下瀏覽 Markdown 檔案，且能記錄觀看的 Markdown 檔案、存入資料庫做為備份。

Markdown Browser 還有下載功能，可用於備份 HackMd 中的 md 檔案

註：紫色部分為新添加的內容

註：此次有使用到的後半課程技術為爬蟲，具體為，會自動爬取 md 中，放在網路的圖片，以及可以從 HackMd 中下載 md 檔案

# 目錄

摘要 .....	1
目錄 .....	2
1. 題目發想 .....	3
2. 功能說明 .....	3
3. 實作技術 .....	3
4. 成品功能描述 .....	6
5. 自我評估 .....	6

## 1. 題目發想

有感於在不將 md 檔案上傳至 HackMd 或者 GitHub 的情況下，只能閱讀最原始的 md 語法，且對於本地的 md 檔案，若存在圖片會導致檔案管理的困難。因此開發了 Markdown Browser。

且由於自身在使用 HackMd 時，時常感到 HackMd 沒有備份功能而沒有安全感，因此添加了爬蟲功能，結合原本的瀏覽功能以及資料庫，可以備份 md 檔案的同時，在本地進行瀏覽。

## 2. 功能說明

### 1. 匯入 md 檔案

可以從本地選取並匯入 md 檔案

### 2. 紀錄 md 檔案

Markdown Browser 會記錄匯入過的檔案並存入資料庫，可以供日後讀取以及作為備份

### 3. 下載 md 檔案

可以輸入開放未登入者瀏覽的 HackMd 網址，將網頁中的內容下載成 md 檔案

### 4. 編輯 md 檔案

可以修改存入資料庫中的 md 檔案的內容

### 5. 匯出 md 檔案

可以將資料庫中的內容再匯出到指定的位置

## 3. 實作技術

Markdown Browser 是以 python 中的 flask 作為框架，搭配 html、CSS、JavaScript 以及 SQLite 完成。以 flask 進行路由，回傳 html、接收 POST 資料。

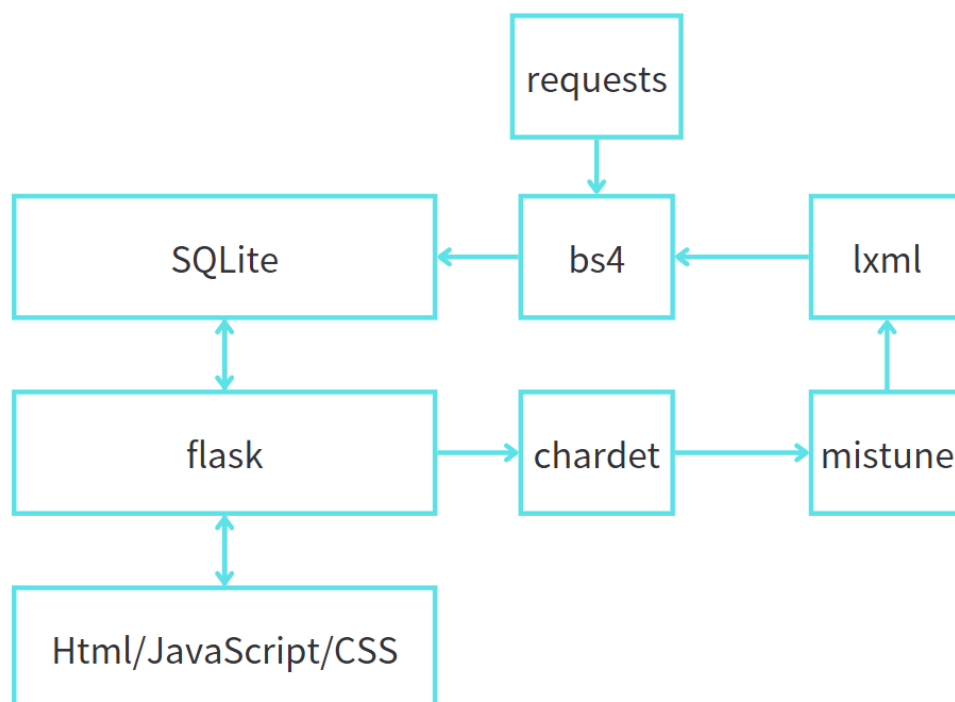
讀取 md 檔案時，首先會篩選 md 檔案中存在本地的圖片，讀取後轉成 Base64，在利用 mistune 套件將 md 檔案轉成 html 格式後，會生成對應的 JavaScript 內容，將指定的圖片路徑刪除，並指定對應圖片的 Base64 內容為

src。並利用 BeautifulSoup4 將生成的 html 放入準備好的 html 模板中，最後將生成的 html 存入資料庫的同時，藉由 flask 回傳 html。

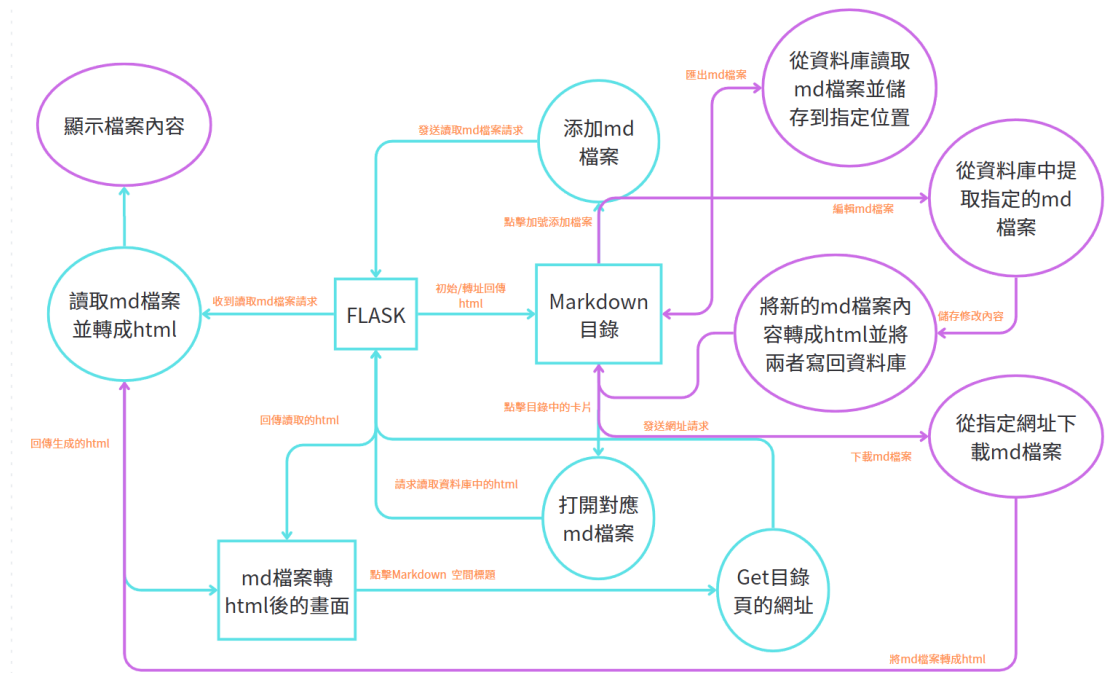
對於讀取過的 md 檔案，會被存入資料庫（包含依據時間生成的 ID、md 檔案的名稱、轉成的 html、md 程式碼、md 檔案中所有圖片轉成的 base64）中。每次回到目錄時，會讀取資料庫中儲存的檔案，並在目錄形成對應的卡片。可以藉由點擊卡片觸發讀取資料庫中對應的檔案，再藉由 flask 回傳 html 檔案，達到開啟檔案的效果，同時利用 CSS 調整介面的顯形式達到美觀效果。

另外，Markdown Browser 中添加了爬蟲功能、編輯功能以及匯出功能。爬蟲功能藉由 requests 套件以及 bs4 實作，可以前往指定網址的 HackMd，下載其中的 md 程式碼，並且會爬取 md 程式碼中的圖片的網址並下載圖片到本地。編輯功能則藉由 SQLite 的資料庫實作達成。由於在添加新的 md 檔案時，會同時將 md 檔案的內容以及其中的所有圖片存入資料庫中，因此在編輯 md 檔案時可以藉由該檔案的 ID 直接從資料庫中讀取其 md 程式碼，修改完成後則會進行與添加 md 檔案相同的步驟，將新的內容轉成 html 後再更新回資料庫。最後的匯出功能也是藉由 SQLite 的資料庫完成，藉由指定 md 檔案的 ID 從資料庫中提取 md 程式碼以及其中的所有圖片的 Base64，並將它們寫入到指定的位置。

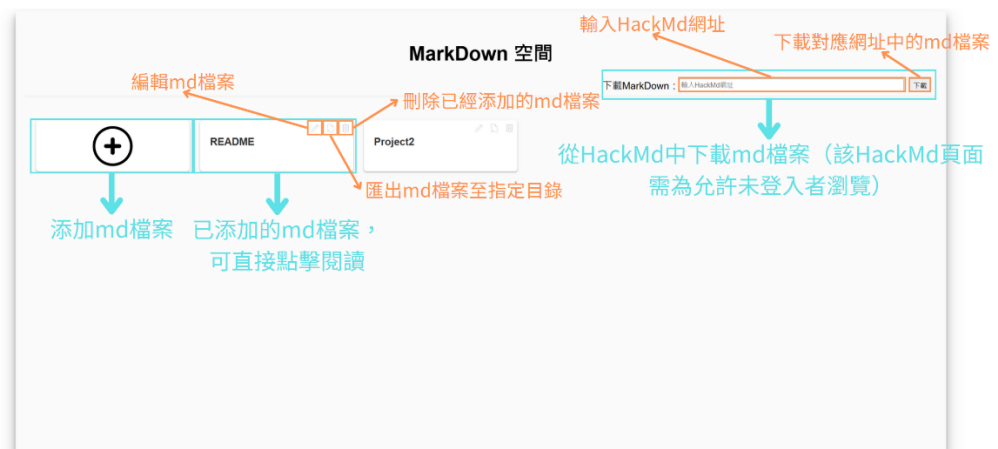
### 架構圖



## 流程圖



## 目錄



## 瀏覽介面



## 4. 成品功能描述

Markdown Browser 的首頁即是儲存的 md 檔案的目錄。在目錄處可以選擇匯入新的 md 檔案，程式會讀取該 md 檔案，生成 html 並存入資料庫後，將使用者導向生成的 html 頁面。在瀏覽 md 檔案（此時已轉成 html）的時候可點擊上方的標題回到目錄

在匯入 md 檔案後，會在目錄頁生成相關卡片，此後可以點擊該卡片打開對應的檔案進行閱讀。

在目錄中有輸入框，可以輸入 HackMd 的網址，下載其中的內容成為 md 檔案

在卡片中，有編輯選項，可以修改指定的 md 檔案。匯出選項，可以將指定的 md 檔案匯出、刪除選項，刪除指定 md 檔案。

## 5. 自我評估

我個人對於此份專題還算滿意，在設計的時候構想的功能基本上都實現

了。唯一比較可惜的是，在設計的最初，**Markdown Browser** 應該要能達到 **HackMd** 那般的美工效果。但是因為我只找到最基礎得的將 **md** 轉成 **html** 的套件，能達到專案此時效果還是我在之後對轉成的 **html** 內容進行二次設計。所以想要達到 **HackMd** 那般的效果可能需要自己寫一個直譯器，或者嘗試將 **md** 內容上傳到 **HackMd**，再下載生成的 **html** 內容。不過因為前者需要花費大量的時間，後者感覺不倫不類，再加上二次設計後的美工效果已經達到和 **GitHub** 上觀看差不多的程度，因此我就沒有繼續改動。

此次為 **Markdown Browser** 添加了爬蟲功能，利用爬蟲技術爬取 **HackMd** 上開放存取的 **md** 內容，結合當前 **Markdown Browser** 的功能，就形成了 **HackMd** 備份功能。再加上添加的匯出、編輯功能，讓 **Markdown Browser** 擁有一套完整的邏輯，可以從本地或者 **HackMd** 備份 **md** 檔案到資料庫，並利用 **Markdown Browser** 進行閱讀。存入資料庫後還可以進行修改。需要時還可以重新重資料庫中匯出成 **md** 檔案。

最後，我認為 **Markdown Browser** 未來可以加強的部分，是編輯功能。此次編輯功能的時做只是簡單的將 **md** 檔案的內容放到一個文字框中，待到使用者儲存時將文字框的內容再次轉成 **html** 後更新到資料庫。但是編輯 **md** 檔案的過程並不 **user friendly**，全程都是如記事本般的存文字，而不像是在 **Visual Studio Code** 或者 **HackMd** 上編寫 **md** 檔案時會有各種顏色進時提示。因此，未來或許可以嘗試添加此類功能，讓編輯時更加簡便。