

Document

1. An introduction of your application, including why you want to develop the application and the main functions of your application.

- Motivation

在這個資訊科技快速發展的世代，對於交易的方式早已不僅限於現金交易。隨著網路應用的普及，越來越多種不同的貨幣流通於市面上。近幾年來，有鑒於虛擬貨幣和加密貨幣的蓬勃發展，引起了我們的興趣。這些貨幣就有如股票、期貨等投資工具，每日都會隨著整個經濟市場而產生變動，要隨時注意這些幣值的交易量及變化，又或是它是否擁有深厚的基本盤，可以穩定的確保收益。在這個匆忙的時代，我們多半沒有那麼充裕的時間去顧及這門課題。所以我們希望能夠透過所學的資料庫技巧，進行快速的數據收集及分析，使我們可以便捷的運用這個工具，去了解這些貨幣市值的變化。讓大家可以在進行繁忙的事務時，仍舊可以依靠這項工具去評估和增加自己的收益。


- Brief Introduction

我們的應用程式具備了以下功能。

1. 資料庫自動更新 (使用者可關閉此項功能)
2. 查詢加密貨幣資料 (如下所示)，並以表格 (使用者可選擇排序依據)、蠟燭圖或折線圖呈現
 - 幣種
 - 日期
 - 最高價
 - 最低價
 - 開盤價
 - 收盤價
 - 交易量
 - 漲跌幅

2. Database design - describe the schema of all your tables in the database, including keys and index, if applicable (why you need the keys, or why you think that adding an index is or is not helpful).

- schema & keys



public
cryptocurrency
type character varying(256)
date date
closing_price double precision
opening_price double precision
highest_price double precision
lowest_price double precision
volume double precision
change double precision

- constraints

- 收盤價 closing_price ≥ 0
- 開盤價 opening_price ≥ 0
- 最高價 highest_price ≥ 0
- 最低價 lowest_price ≥ 0
- 交易量 volume ≥ 0

- index

我們的 database 中沒有額外的 index。多數人查詢歷史資料不會單看一項指標，而是著眼隨著時間推移，各指標的變化。而 {幣種, 日期} 作為 primary key，DBMS 會自動為其建立 index，因此不需要再額外為日期建立 index。

3. Database design - describe the normal form of all your tables. If the tables are not in BCNF, please include the reason for it (performance trade-off, etc.).

我們的 database 裡只有一個 table : cryptocurrency。這個 table 遵從 BCNF，因為開盤價、最高價、最低價、收盤價、交易量、漲跌幅之間並無對應關係，故彼此間無 functional dependency。因此僅有的 non-trivial functional dependency 為

{幣種, 日期} -> {收盤價, 開盤價, 最高價, 最低價, 交易量}
而 {幣種, 日期} 是 super key，因此這個 table 遵從 BCNF。

4. From the data sources to the database - describe the data source and the original format.

- data source : **investing.com**

investing.com 上有各個幣種的資訊並且這些資訊也符合我們 application 所需並且他資料的型態也和我們 database 的儲存方式相近，還能提供csv 檔下載的功能以及該網站也會更新不同幣值的變動，適合我們當作資料的來源。

- original format :

以Bitcoin作為範例，資料依序為 {日期, 收盤價, 開盤價, 最高價, 最低價, 交易量, 漲跌幅}

"Jan 04,
2023", "16,862.40", "16,674.20", "16,890.90",
"16,656.50", "209.81K", "1.13%"

5. From the data sources to the database - describe the methods of importing the original data to your database and strategies for updating the data, if you have one.

- importing original data

手動下載 (investing.com 有提供歷年資料下載)，再用 pgAdmin4 import data

- original format :

csv 檔，其中的一行如下

```
"Jan 04,  
2023", "16,862.40", "16,674.20", "16,890.90",  
"16,656.50", "209.81K", "1.13%"
```

- 透過 validate_and_merge_csv.exe 將格式改成可以用 pgAdmin4 import 的 csv 檔。source code 是 validate_and_merge_csv.cpp。

- strategies for updating

用爬蟲取得最新 data，並透過 Python 的 psycopg 套件將 data insert into database。

- 透過 web_crawler2.py 從網頁獲取 data。
 - 透過 **validate.py** 更改 data 格式。
 - 透過 **Updater.py** 將 data insert into database。
data 更新時機
 - 有使用者希望取得最新 data 時，會對 database 進行 query，查看是否已含最新的 data。若 database 中無最新 data 時即更新。

6. Application with database - explain why your application needs a database.

若想以原網站閱覽多種加密貨幣的資料，就需要開啟許多網頁。因此我們希望能開發出一個軟體，選好幣種與日期區間後就能直接取得資料，免除開啟許多網頁的麻煩。由於資料量很大，僅 20 種加密貨幣的資料就已經有數萬筆，因此需要資料庫才能進行有效率的查詢與管理，並方便進一步開發。未來可開發方向如下：

- 趨勢預測
- 即時價格查詢
- 多幣種趨勢比較

7. Application with database - includes the queries that are performed by your application, how your application performed these queries (connections between application and database), and what is the cooperating functions for your application.

- 主資料庫架設在 AWS 的 RDS 上，採用 PostgreSQL 14 版。主程式以 python 編寫，使用者介面採 PyQt6 套件，而 psycopg 套件則是負責處理 python 上的 PostgreSQL 操作。

- getCryptoType

```
SELECT DISTINCT type
FROM cryptocurrency
ORDER BY type
```

- getHistoricalData

```
SELECT *
FROM cryptocurrency
WHERE
    type = '{cname}' AND
    date BETWEEN '{sdate}' AND '{edate}'
ORDER BY {order} {dir}
```

{cname}、{sdate}、{edate}、{order}、{dir} 為 Python 主程式中的 string variable，可供使用者決定下列幾項

- 幣種
 - 日期區間
 - 排序依據
 - 升冪/降冪排序
- update
- 這部分在第 5 題中已經有相關解釋，不再贅述。

8. All the other details of your application that you want us to know.

- graphing

將獲取的資料繪製成 K 線圖，細線的最高與最低點分別代表當日的最高與最低價，粗線的最高與最低點則代表當日的開盤以及收盤價。不同的顏色則代表當日該幣種開盤與收盤價間的關係，若開盤價大於收盤價繪製陰線 (綠色)，收盤價大於開盤價繪製陽線 (紅色)

。

- Further Improvements : Local SQL-statements

所有 SQL 操作皆是由用戶端操作，包括 CREATE TABLE、INSERT、SELECT，因此會有惡意攻擊資料庫的問題、像是 SQL Injection 等。所有操作都應該由 AWS 方提供以防安全漏洞。