國立臺北商業大學

資 訊 管 理 系

113’資訊系統專案設計

**系統手冊**

一張含有 圖表, 設計 的圖片

自動產生的描述

**組 別：第113502組**

**題 目： CodeAlchemy系統**

**指導老師：楊進雄老師**

**組 長：10956010 童柏祐**

**組 員：10956011 周恩宇** **10956030 林妤玟**

**10956044 顏子綺** **10956051 范雨茜**

**中 華 民 國 1 1 3 年 5 月 2 7 日**

**目錄**

[第一章　前言 1](#_Toc179183526)

[1-1 背景簡介 1](#_Toc179183527)

[1-2 動機 1](#_Toc179183528)

[1-3 系統目的與目標 2](#_Toc179183529)

[1-4 預期成果 2](#_Toc179183530)

[第二章 營運計畫 3](#_Toc179183531)

[2-1 可行性分析 3](#_Toc179183532)

[2-2 商業模式 3](#_Toc179183533)

[2-3 市場分析-STP 3](#_Toc179183534)

[2-4 競爭力分析 SWOT-TOWS 4](#_Toc179183535)

[第三章 系統規格 5](#_Toc179183536)

[3-1系統架構 5](#_Toc179183537)

[3-2系統開發工具 5](#_Toc179183538)

[3-3系統使用環境開發工具 6](#_Toc179183539)

[第四章 專案時程與組織分工 7](#_Toc179183540)

[4-1專案時程圖 7](#_Toc179183541)

[4-2 專案組織與分工 8](#_Toc179183542)

[第五章 需求模型 11](#_Toc179183543)

[5-2 需求清單 12](#_Toc179183544)

[第六章 程序模型 13](#_Toc179183545)

[6-1 資料流程圖(Data flow diagram) 13](#_Toc179183546)

[6-2程序規格書(Process specification) 18](#_Toc179183547)

[第七章 資料模型 23](#_Toc179183548)

[7-1 實體關聯圖(Entity relationship diagram) 23](#_Toc179183549)

[7-2資料字典(Data dictionary) 24](#_Toc179183550)

[第八章 資料庫設計 25](#_Toc179183551)

[8-1資料庫關聯表(Constraints) 25](#_Toc179183552)

[8-2表格及其Meta data 26](#_Toc179183553)

[第九章 程式 30](#_Toc179183554)

[9-1 軟體架構與程式清單 30](#_Toc179183555)

[9-2程式規格描述 30](#_Toc179183556)

[第十章 測試模型 71](#_Toc179183557)

[10-1 測試計畫 71](#_Toc179183558)

[10-2 測試個案與測試結果 72](#_Toc179183559)

[第十一章 操作手冊 76](#_Toc179183560)

[第十二章 使用手冊 77](#_Toc179183561)

[第十三章 心得感想 78](#_Toc179183562)

[第十四章 參考資料 79](#_Toc179183563)

[附錄一 評審建議與回應 80](#_Toc179183564)

**圖目錄**

[圖 3-1-1 系統架構 5](#_Toc167026211)

[圖 4-1-1 專案時程 7](#_Toc167026212)

[圖 5-1-1 功能分解圖 11](#_Toc167026213)

[圖 6-1-1 系統環境圖 13](#_Toc167026214)

[圖 6-1-2 DFD圖0 14](#_Toc167026215)

[圖 6-1-3 DFD圖1 15](#_Toc167026216)

[圖 6-1-4 DFD圖2 15](#_Toc167026217)

[圖 6-1-5 DFD圖3 16](#_Toc167026218)

[圖 6-1-6 DFD圖4 16](#_Toc167026219)

[圖 6-1-7 DFD圖5 17](#_Toc167026220)

[圖 6-1-8 DFD圖6 17](#_Toc167026221)

[圖 6-1-9 DFD圖7 **錯誤! 尚未定義書籤。**](#_Toc167026222)

[圖 6-1-10 DFD圖8 **錯誤! 尚未定義書籤。**](#_Toc167026223)

[圖 7-1-1 實體關聯圖 23](#_Toc167026224)

[圖 8-1-1 資料庫關聯圖 25](#_Toc167026225)

**表目錄**

[表 2-4-1 SWOT分析 4](#_Toc167023861)

[表 2-4-2 TOWS分析 4](#_Toc167023862)

[表 3-2-1 系統軟、硬體需求 5](#_Toc167023863)

[表 3-3-1 開發工具 6](#_Toc167023864)

[表 4-2-1 專案組織分工表 8](#_Toc167023865)

[表 4-2-2 專題成果工作內容與貢獻度表 10](#_Toc167023866)

[表 5-2-1 功能性需求 12](#_Toc167023867)

[表 5-2-2 非功能性需求 12](#_Toc167023868)

[表 6-2-1 程序規格書-題目作答 18](#_Toc167023869)

[表 6-2-2 程序規格書-收藏題目 18](#_Toc167023870)

[表 6-2-3 程序規格書-搜尋題目 18](#_Toc167023871)

[表 6-2-4 程序規格書-新增討論 **錯誤! 尚未定義書籤。**](#_Toc167023872)

[表 6-2-5 程序規格書-修改討論 **錯誤! 尚未定義書籤。**](#_Toc167023873)

[表 6-2-6 程序規格書-刪除討論 **錯誤! 尚未定義書籤。**](#_Toc167023874)

[表 6-2-7 程序規格書-創建比賽 19](#_Toc167023875)

[表 6-2-8 程序規格書-加入比賽 19](#_Toc167023876)

[表 6-2-9 程序規格書-新增會員 19](#_Toc167023877)

[表 6-2-10 程序規格書-會員資料 20](#_Toc167023878)

[表 6-2-11 程序規格書-新增反饋 20](#_Toc167023879)

[表 6-2-12 程序規格書-反饋結果 20](#_Toc167023880)

[表 6-2-13 程序規格書-新增題目 21](#_Toc167023881)

[表 6-2-14 程序規格書-修改題目 21](#_Toc167023882)

[表 6-2-15 程序規格書-刪除題目 21](#_Toc167023883)

[表 6-2-16 程序規格書-查詢題目 22](#_Toc167023884)

[表 6-2-17 程序規格書-新增討論群 **錯誤! 尚未定義書籤。**](#_Toc167023885)

[表 6-2-18 程序規格書-修改討論群 **錯誤! 尚未定義書籤。**](#_Toc167023886)

[表 6-2-19 程序規格書-刪除討論群 **錯誤! 尚未定義書籤。**](#_Toc167023887)

[表 6-2-20 程序規格書-反饋結果 22](#_Toc167023888)

[表 6-2-21 程序規格書-反饋內容 22](#_Toc167023889)

[表 7-2-1 資料字典 24](#_Toc167023890)

[表 8-2-1 user資料表 26](#_Toc167023891)

[表 8-2-2 feedback資料表 26](#_Toc167023892)

[表 8-2-3 discussion資料表 **錯誤! 尚未定義書籤。**](#_Toc167023893)

[表 8-2-4 problem資料表 27](#_Toc167023894)

[表 8-2-5 answer record資料表 28](#_Toc167023895)

[表 8-2-6 collection資料表 28](#_Toc167023896)

[表 8-2-7 contest problem資料表 28](#_Toc167023897)

[表 8-2-8 contest資料表 29](#_Toc167023898)

[表 8-2-9 contest participant 資料表 29](#_Toc167023899)

第一章　前言

**1-1 背景簡介**

隨著數位時代的來臨，程式設計成為了現代社會不可或缺的一部分，對於學生、工程師和技術人員而言，具備良好的程式設計基礎和解題能力已經成為進入高科技行業和取得成功的必要條件之一。

在過去的學習和實踐過程中，我們曾花費大量的時間在尋找高質量的學習資源上。我們意識到現有的學習平台雖然提供了大量的題目和教材，但傳統的教育方式往往缺乏趣味性和互動性，使學生們難以找到與之相符的學習方法。

Dolos系統，是一個先進的程式碼比對工具，能夠在學術和競賽環境中檢測程式碼的相似性。同時還具備著程式碼優化上的建議、個人風格檢查與錯誤檢測功能，能幫助學習者不僅能尋找快速又簡單的方法破解程式題目，更能提升自身的程式品質與實戰能力。

**1-2 動機**

我們的動機源於對技術教育的貢獻渴望，希望能打造一個開放、包容、高效的程式設計學習平台，為所有程式設計學習者提供一個共同成長的空間。我們希望通過CodeAlchemy這個平台，解決現有程式設計學習資源的種種問題，讓每一位學習者都能夠輕鬆、有趣地學習程式設計，提升自己的程式技術水平。

**1-3 系統目的與目標**

1. 提供更多樣的學習資源：提供廣泛且高品質的程式設計學習資源，包括教材、練習題、項目案例等，以幫助使用者們掌握各種程式語言和技術。

2. 執行多種語言：系統可以同時支援C/C++、JAVA、Python、JavaScript...等多種語言。

3. 技術交流和創新：促進技術交流和創新，鼓勵學生參與共同開發項目、參加技術競賽等活動，從而激發創新潛力和培養團隊合作能力。

4. 防止潛在抄襲行為：使用Dolos系統來檢測使用者提交的程式，辨別是否存在重複性或潛在抄襲行為，同時了解程式碼中存在的相似性，促進使用者反思自己寫程式的習慣，從而提升程式的創新性與質量。

**1-4 預期成果**

本專題主要建立一個全面的學習資源平台，該平台將提供各種程式語言和技術的高品質教材和練習題，讓學習者能夠提升程式能力，並透過系統來創新出更多不同的程式解法。在技術上的實踐，激勵各種學生們參與創新項目和解決實際問題，從而促進技術創新和人才培養，為技術社群的發展注入新的活力和動力。

第二章 營運計畫

**2-1 可行性分析**

1. 程式碼相似度比對系統是一個幫助開發人員檢視代碼重複的程度、尋找代碼抄襲的功用。

2. 技術可行性評估：我們評估了 Dolos、Diff Checker、CodePen 和 pycode similar 這些現有的程式碼比對工具。由於每個工具產生的比對結果有時都有所不同，因此我們採取使用率較高的工具來產生最終結果。

3. 經濟可行性評估：需考慮系統開發和維護階段的成本，包括技術人員、資料庫和設備方面的費用。此外，還需考量系統的行銷成本，提升CodeAlchemy在市場上的知名度和使用率。值得一提的是，相關產品Codequiry和Copylacks兩個成功商品化的典型案例，將為本系統提供了寶貴的參考和啟示，以確保系統的成功發展和長期可持續性。

4. 風險評估：在開發過程中，主要面臨技術層面的挑戰，若是提供服務遜色於同類型的系統，將對我們構成競爭壓力。因此，系統需要不斷地提升技術水平、追蹤行業動態，以及保持創新和敏捷，以應對競爭帶來的挑戰，確保能在市場上的領先地位。

**2-2 商業模式**

1. 定位：目標是協助開發人員在開發過程中提升程式執行效能，即降低程式碼的重複程度。同時，CodeAlchemy也能夠確認程式碼是否存在抄襲疑慮，以確保程式碼的原創性。有助於提高程式碼的品質和可靠性。

2. 教育與學術市場模式：基於Dolos檢查抄襲的功能，應用於學術課堂。還可以透過分析程式碼的結構與效率來了解學生在程式方面的學習能力，同時盡可能預防學生的作弊行為。此外，根據學生的程式碼提交情況和比對結果，來分析他們的學習進度與成長曲線，並提供具體的反饋，讓學習者了解自己在哪些方面需要進一步提升。

**2-3 市場分析-STP**

1. 市場區隔：教育機構市場，提供管理課程、教學資源和分析學生學習情況的工具。

2. 市場目標：主要為學校、大學、培訓中心和初學程式設計者。

3. 市場定位：網站提供的易用性、互動性和豐富的教育資源，以幫助學生更好地學習和提高技術水平。

**2-4 競爭力分析 SWOT-TOWS**

▼表2-4-1 SWOT分析

|  |  |
| --- | --- |
| 優勢(Strengths) | 弱勢(Weaknesses) |
| 1. 提供中文題庫，讓看不懂英文的人也能輕易讀懂題目。 2. 提供了題目解題影片，當使用者解題時對題目毫無頭緒時可以從解題影片中尋求解法。 3. 利用Dolos系統了解程式中的相似性，讓使用者反思寫程式時的習慣，從而提升程式的創新性。 | 1. 要持續開發和維護一個高質量的教育平台需要大量的技術和資源。 2. 無法及時調整和改進平台，可能會影響使用者的滿意度。 3. 系統無法處理某些程式語言，需要進一步改善。 |
| 機會(Opportunities) | 威脅(Threats) |
| 1. 可以透過不斷更新，以此來精進平台的精準度和性能，提供更加優質的服務。 2. 與教育機構合作，為學生創造新的教育機會，提高學習效率。 | 1. 新技術的出現和技術的變化可能會對平台的技術架構和功能產生影響，需要及時跟進和調整。 2. 程式設計教育市場已經飽和，存在過多的競爭對手和類似的教育平台，使得剛進入市場和吸引用戶變得更加困難。 |

▼表2-4-2 TOWS分析

|  |  |  |
| --- | --- | --- |
|  | S優勢 | W劣勢 |
| O  機會 | 邀請專業團隊來開發新的學習課程或是培訓計畫，來滿足需要更多課程的學生。 | 透過學術機構或是教育機構建立合作夥伴關係，來提供學生學會使用系統，收集更多回饋，以此來改進系統內容。 |
| T  威脅 | 利用中文題庫的優勢來吸引那些不懂英文的用戶，特別是在中文使用者眾多的地區或市場。 | 透過更多教育機構或技術公司合作，來共享資源和技術，以此減輕資源的不足和技術落後所帶來的威脅。 |

第三章 系統規格

**3-1系統架構**

一張含有 圖表, 寫生, 圖畫, 設計 的圖片

自動產生的描述

* 圖 3-1-1 系統架構

**3-2系統開發工具**

▼表3-2-1 系統軟、硬體需求

|  |  |
| --- | --- |
| 軟體需求 | |
| 作業系統需求 | Windows 10(版本1903或更新版本) |
| 硬體需求 | |
| 處理器 | 64位元處理器，最好是支援虛擬化技術的處理器，如Intel VT-x或AMD-V。 |
| 記憶體 | 至少8GB RAM。  建議配置16GB RAM或更高。 |
| 儲存空間 | 至少20GB可用空間。 |
| 虛擬化技術 | 需要啟用虛擬化技術。  可以在BIOS或UEFI設定中啟用它。 |
| 顯示器 | 最佳解析度為1024 x 768或更高。 |

**3-3系統使用環境開發工具**

▼表 3-3-1 開發工具

|  |  |  |
| --- | --- | --- |
| 系統開發環境 | | |
| 作業系統 |  | Windows、Ubuntu |
| 撰寫工具 |  | Visual Studio Code、MySQL workbench |
| 程式開發語言 | | |
| 前端 |  | HTML、Bootstrap、CSS |
| 後端 |  | Python、Flask |
| 資料庫 |  | MySQL |
| 介面及插圖繪製工具 | | |
| 插圖 | Adobe Illustrator - 維基百科，自由的百科全書 | Bootstrap Icons |
| 介面 |  | Figma |
| 文件及美化工具 | | |
| 文件 |  | Microsoft Word、ChatGPT |
| 簡報 |  | Microsoft PowerPoint |
| 專案管理及版本控制工具 | | |
| 應用程式 |  | Visual Studio Code |
| 版本控制 |  | Git |

第四章 專案時程與組織分工

**4-1專案時程圖**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 項目  時間 | 2024 | | | | | | | | | |
| 1月 | 2月 | 3月 | 4月 | 5月 | 6月 | 7月 | 8月 | 9月 | 10月 |
| 主題制訂與討論 |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
| 文件製作 |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
| 資料庫建置 |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
| 前端頁面撰寫 |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
| 後端頁面撰寫 |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
| 介面設計 |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
| LOGO設計 |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
| 系統測試 |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
| 簡報製作 |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
| 影片製作 |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |

* 圖 4-1-1 專案時程

**4-2 專案組織與分工**

▼表 4-2-1 專案組織分工表

●主要負責人 ○次要負責人

| 項目/組員 | | 10956010  童柏祐 | 10956011  周恩宇 | 10956030  林妤玟 | 10956044  顏子綺 | 10956051  范雨茜 |
| --- | --- | --- | --- | --- | --- | --- |
| 後端開發 | 資料庫建置 | ● |  |  | ○ |  |
| 伺服器架設 | ○ | ● |  |  |  |
| 帳號管理 |  | ● |  |  |  |
| 題目管理 |  | ● |  |  |  |
| 比賽管理 |  |  |  | ● |  |
| 會員管理 |  | ● |  |  |  |
| 反饋管理 |  |  |  |  | ● |
| 排行榜 |  |  | ● |  |  |
| 爬蟲 | ● | ○ |  |  |  |
| 前端開發 | 註冊+登入 |  | ● |  |  |  |
| 解題介面 | ○ | ● |  |  |  |
| 系統主畫面 |  | ● |  |  |  |
| 排行榜 |  |  | ● |  |  |
| 創建比賽 |  |  |  | ● |  |
| 加入比賽 |  |  |  | ● |  |
| 會員資料 |  | ● |  |  |  |
| 程式碼分析畫面 |  | ● |  |  |  |
| 反饋 |  |  |  |  | ● |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
| 美術設計 | UI/ UX | ○ | ● |  |  |  |
| Web/APP介面設計 |  | ● |  |  |  |
| 色彩設計 |  |  |  | ● |  |
| Logo設計 |  |  |  | ● |  |
| 素材設計 |  |  |  | ● |  |
| 文件撰寫 | 統整 |  |  | ● | ○ |  |
| 第1章 前言 |  |  | ● |  |  |
| 第2章 營運計畫 |  |  | ● |  |  |
| 第3章 系統規格 |  |  |  | ● |  |
| 第4章 專題時程與組織分工 |  |  | ○ |  | ● |
| 第5章 需求模型 |  |  | ○ | ○ | ● |
| 第6章 程序模型 |  |  | ○ | ○ | ● |
| 第7章 資料模型 |  |  | ○ | ○ | ● |
| 第8章 資料庫設計 | ● |  | ○ |  | ○ |
| 第9章 程式 |  | ● |  |  |  |
| 第10章 測試模型 |  |  | ● |  |  |
| 第11章 操作手冊 |  |  | ● |  |  |
| 第12章 使用手冊 |  |  | ● |  |  |
| 其 他 | 簡報製作 |  |  | ● |  |  |
| 影片製作 |  | ● |  |  |  |

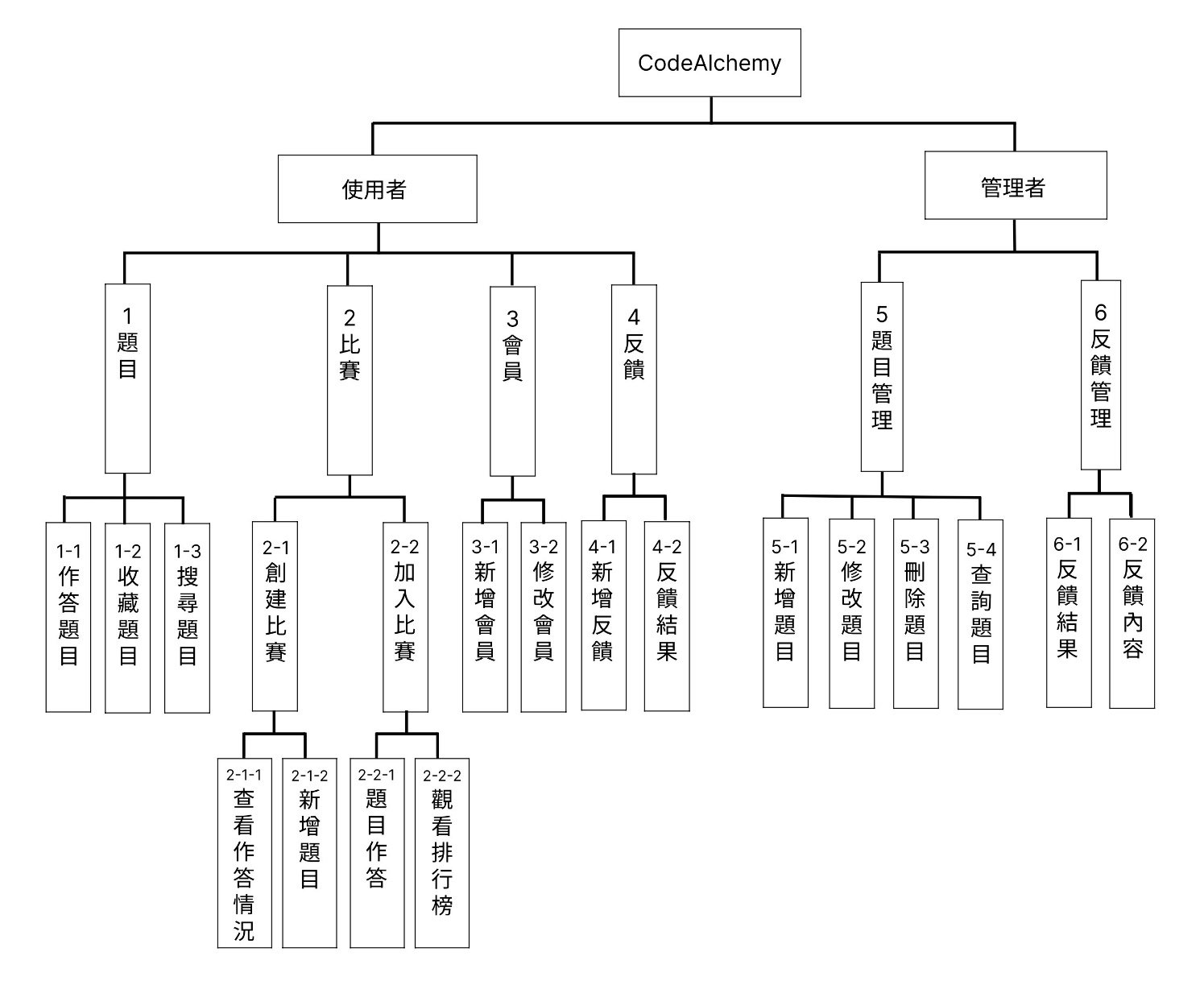
▼表 4-2-2 專題成果工作內容與貢獻度表

|  |  |  |  |
| --- | --- | --- | --- |
| 序號 | 姓名 | 工作內容<各限100字以內> | 貢獻度 |
| 1 | 組長  童柏祐 | 建置資料庫、資料庫關聯圖、資料庫管理圖繪製、自動註冊OJ帳號、抓取OJ的題目、提交使用者程式碼並顯示結果、處理作答紀錄、爬蟲。 | 20 % |
| 2 | 組員  周恩宇 | 負責前端UI設計；logo發想；後端登入功能開發；使用Google帳號登入；串接前端後端及資料庫；使用者權限控管；研究flask-mail、codemirror、docker；dolos架設；串接dolos api。 | 20 % |
| 3 | 組員  林妤玟 | 總體文件統整、前言、SOWT分析、TWOS分析、專題時程與組織分工、需求清單、非需求清單、需求模型、程序模型、資料模型、系統簡介。 | 20 % |
| 4 | 組員  顏子綺 | logo設計、素材設計、系統規格、文件第八章、文件統整、簡報製作、創建比賽介面、創建比賽後端、加入比賽介面、加入比賽後端、系統架構圖、功能分解圖、資料流程圖。 | 20 % |
| 5 | 組員  范雨茜 | 功能分解圖、資料流程圖、實體關聯圖、專題時程繪製，修改資料庫關聯圖，建置資料庫，反饋與反饋管理的前後端程式與頁面撰寫。 | 20 % |
|  | | | 總計:100% |

一張含有 文字, 螢幕擷取畫面, 數字, 字型 的圖片

自動產生的描述

第五章 需求模型

**5-1 功能分解圖(Functional Decomposition Diagram)**

* 圖 5-1-1 功能分解圖

**5-2 需求清單**

▼表5-2-1 功能性需求

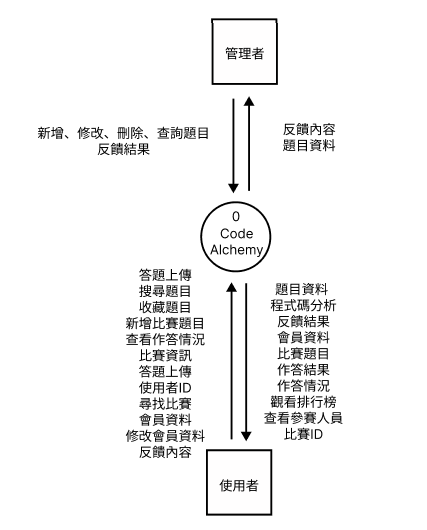
|  |  |  |
| --- | --- | --- |
| **編號** | **功能** | **功能敘述** |
| **1** | 題目 | 1-1作答題目：點擊題目，輸入程式即可上傳。  1-2收藏題目：為使用者儲存喜愛的題目。  1-3搜尋題目：使用者可透過搜尋功能來尋找想要的題目。 |
| **2** | 比賽 | 2-1創建比賽：使用者可創建比賽來考驗使用者的程式語言能力。  2-2加入比賽：可以觀看比賽內容來決定是否參加比賽。 |
| **3** | 會員 | 3-1新增會員：點擊”註冊”，使用者可以新增會員資料到系統內。  3-2修改會員：點擊”個人資料”即可查看使用者存到系統內的個人基本資料。 |
| **4** | 反饋 | 提供使用者來對系統意見反饋。 |
| **5** | 題目管理 | 5-1新增題目：管理者可以新增題目至系統內。  5-2修改題目：管理者可修改系統內的題目內容。  5-3刪除題目：管理者可以刪除題目。  5-4查詢題目：讓管理者更快速查詢到題目。 |
| **6** | 反饋管理 | 提供管理者來觀看使用者所提出意見，並加以改善系統。 |

▼表 5-2-2 非功能性需求

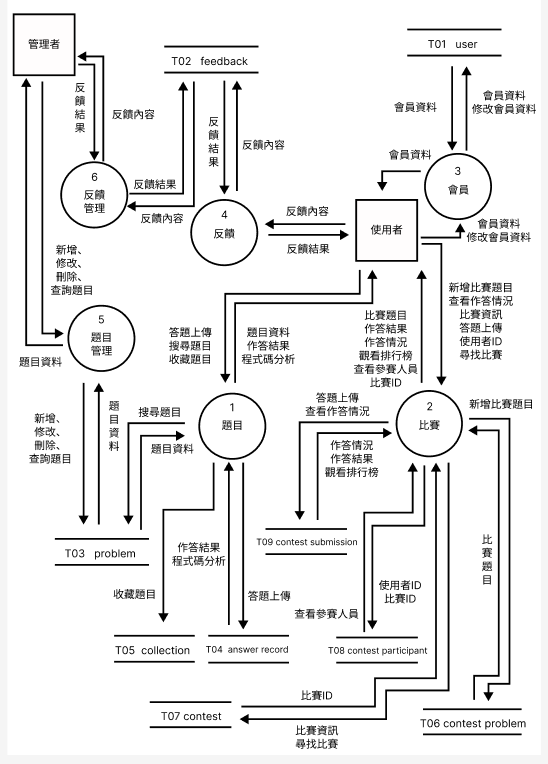
|  |  |
| --- | --- |
| **功能** | **功能敘述** |
| 效能需求 | 確保系統能夠快速、穩定地處理大量的用戶請求和數據操作。 最大限度地優化系統性能，盡量減少頁面加載時間和服務器延遲。 |
| 可用性 | 提供直覺友好的用戶界面，讓使用者能夠輕鬆找到所需要的功能並進行操作。 |
| 穩定性 | 測試系統的穩定性和可靠性，確保在高流量和高負載情況下仍能正常運行。 |

第六章 程序模型

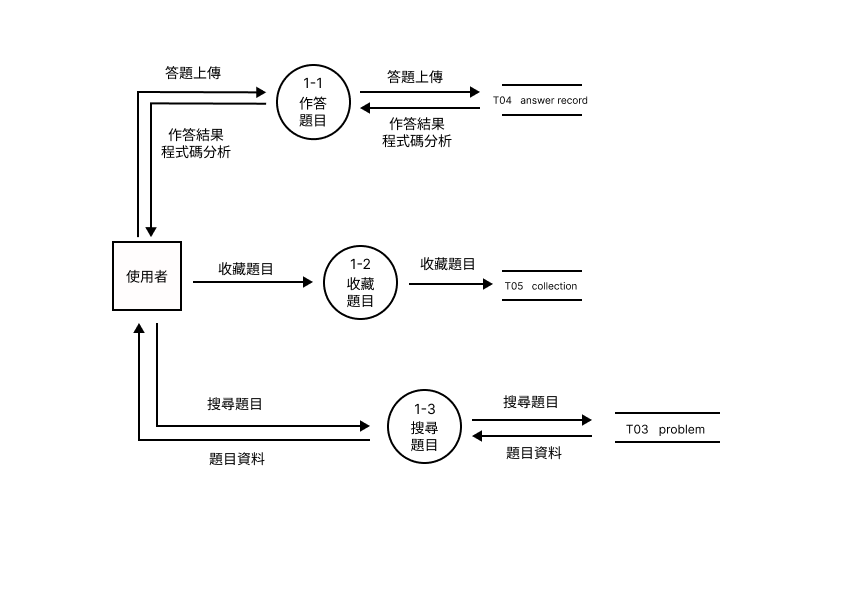
**6-1 資料流程圖(Data flow diagram)**



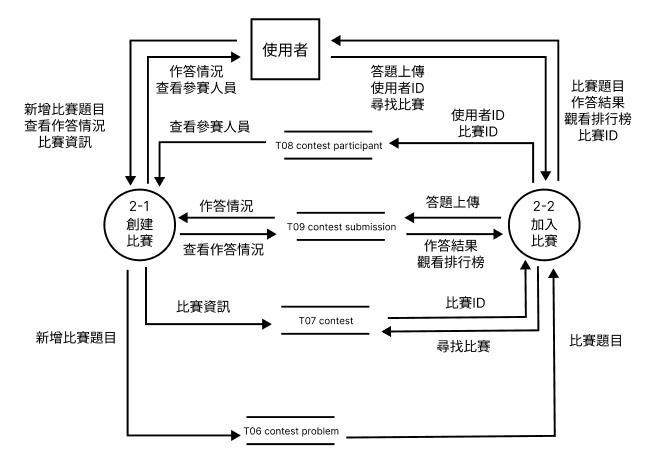
* 圖 6-1-1 系統環境圖



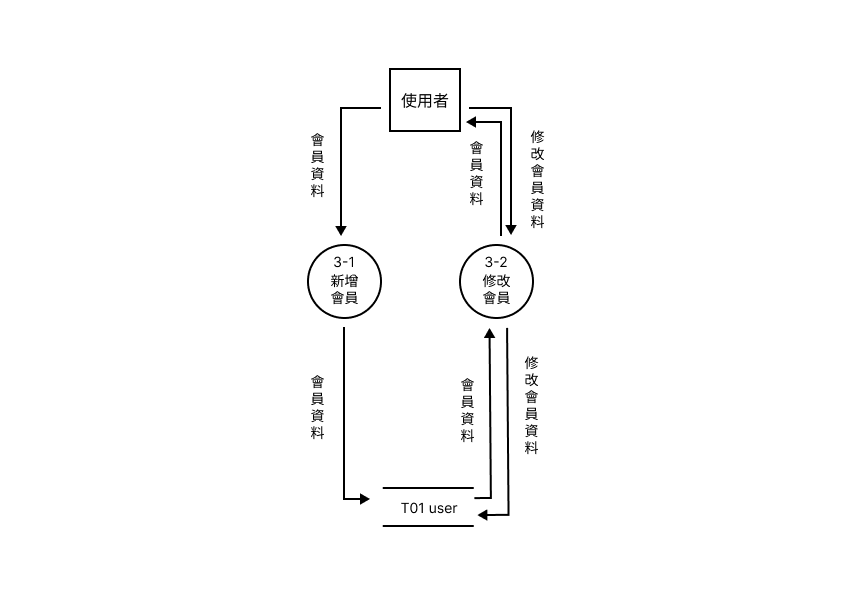
* 圖 6-1-2 DFD圖0



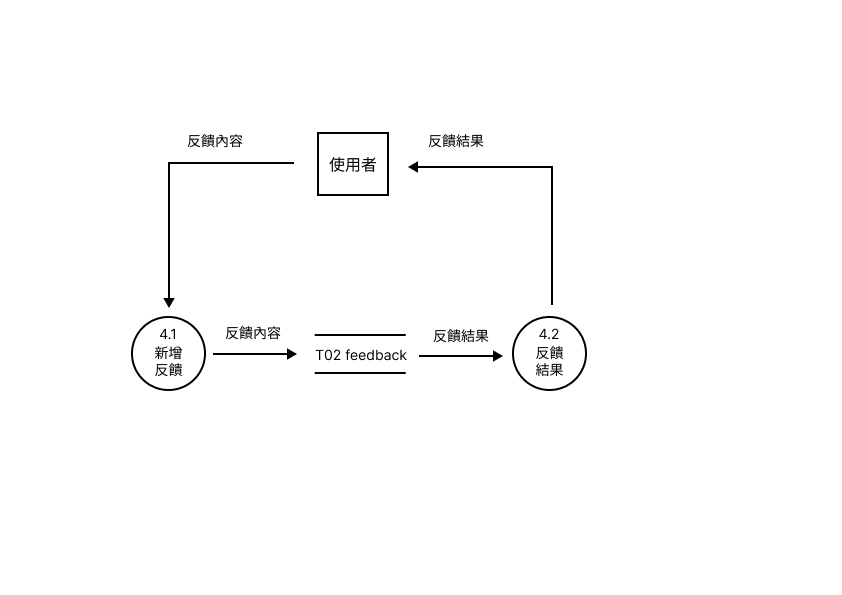
* 圖 1-1-3 DFD圖1



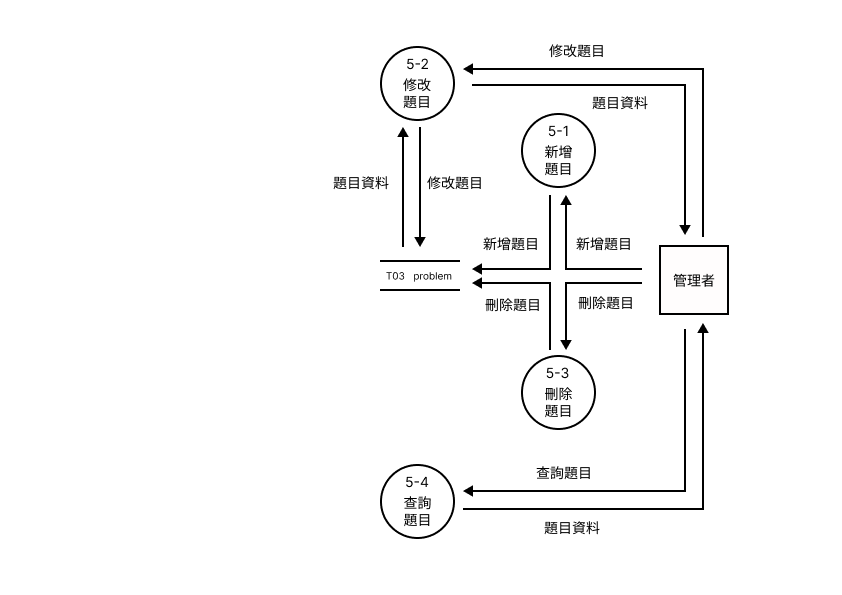
* 圖 6-1-4 DFD圖2



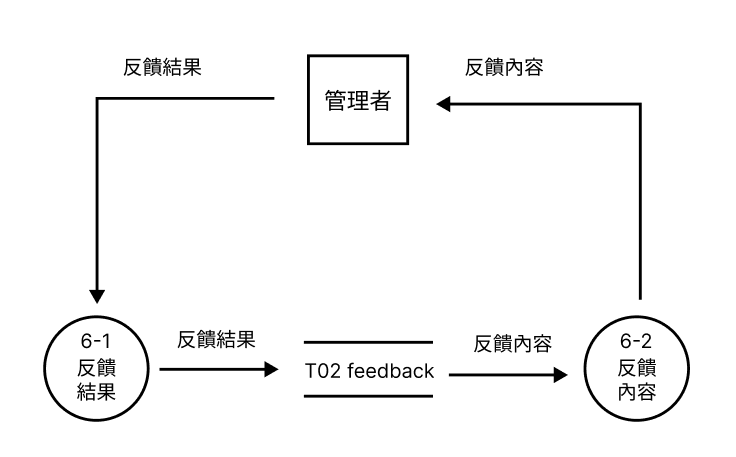
* 圖 6-1-5 DFD圖3



* 圖 6-1-6 DFD圖4



* 圖 6-1-7 DFD圖5



* 圖 6-1-8 DFD圖6

**6-2程序規格書(Process specification)**

▼表6-2-1 程序規格書-題目作答

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-1 | 功能名稱 | 作答題目 |
| 功能說明 | 點擊題目，輸入程式即可上傳 | | |
| 輸入值 | 程式碼 | | |
| 輸出值 | 作答結果和程式碼分析 | | |

▼表 6-2-2 程序規格書-收藏題目

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-2 | 功能名稱 | 收藏題目 |
| 功能說明 | 為使用者儲存喜愛的題目 | | |
| 輸入值 | 欲新增收藏題目資料 | | |
| 輸出值 | 新增成功訊息 | | |

▼表 6-2-3 程序規格書-搜尋題目

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 1-3 | 功能名稱 | 搜尋題目 |
| 功能說明 | 使用者透過搜尋功能來尋找想要的題目 | | |
| 輸入值 | 題目名稱 | | |
| 輸出值 | 查詢成功訊息 | | |

▼表 6-2-4 程序規格書-創建比賽

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 2-1 | 功能名稱 | 創建比賽 |
| 功能說明 | 使用者可創建比賽來考驗使用者的程式語言能力 | | |
| 輸入值 | 新增比賽資料 | | |
| 輸出值 | 新增成功訊息 | | |

▼表 6-2-5 程序規格書-加入比賽

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 2-2 | 功能名稱 | 加入比賽 |
| 功能說明 | 可以觀看比賽內容來決定是否參加比賽 | | |
| 輸入值 | 是否參加比賽 | | |
| 輸出值 | 參加比賽成功訊息 | | |

▼表 6-2-6 程序規格書-新增會員

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 3-1 | 功能名稱 | 新增會員 |
| 功能說明 | 點擊註冊，使用者可以新增會員資料到系統內 | | |
| 輸入值 | 欲註冊資料 | | |
| 輸出值 | 註冊成功訊息 | | |

▼表 6-2-7 程序規格書-會員資料

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 3-2 | 功能名稱 | 修改會員 |
| 功能說明 | 點擊”修改會員資料”即可修改個人資料 | | |
| 輸入值 | 個人資料 | | |
| 輸出值 | 個人資料資訊 | | |

▼表 6-2-8 程序規格書-新增反饋

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 4-1 | 功能名稱 | 新增反饋 |
| 功能說明 | 新增系統反饋資料 | | |
| 輸入值 | 反饋內容 | | |
| 輸出值 | 反饋成功訊息 | | |

▼表 6-2-9 程序規格書-反饋結果

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 4-2 | 功能名稱 | 反饋結果 |
| 功能說明 | 查看反饋結果 | | |
| 輸入值 | 反饋結果 | | |
| 輸出值 | 反饋結果內容 | | |

▼表 6-2-10 程序規格書-新增題目

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 5-1 | 功能名稱 | 新增題目 |
| 功能說明 | 管理者可以新增題目至系統內 | | |
| 輸入值 | 題目標題與題目內容 | | |
| 輸出值 | 新增成功訊息 | | |

▼表 6-2-11 程序規格書-修改題目

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 5-2 | 功能名稱 | 修改題目 |
| 功能說明 | 管理者可修改系統內的題目內容 | | |
| 輸入值 | 修改題目內容 | | |
| 輸出值 | 修改成功訊息 | | |

▼表 6-2-12 程序規格書-刪除題目

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 5-3 | 功能名稱 | 刪除題目 |
| 功能說明 | 管理者可以刪除題目 | | |
| 輸入值 | 刪除題目內容 | | |
| 輸出值 | 刪除成功訊息 | | |

▼表 6-2-13 程序規格書-查詢題目

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 5-4 | 功能名稱 | 查詢題目 |
| 功能說明 | 讓管理者更快速查詢到題目 | | |
| 輸入值 | 題目名稱 | | |
| 輸出值 | 查詢成功訊息 | | |

▼表 6-2-14 程序規格書-反饋結果

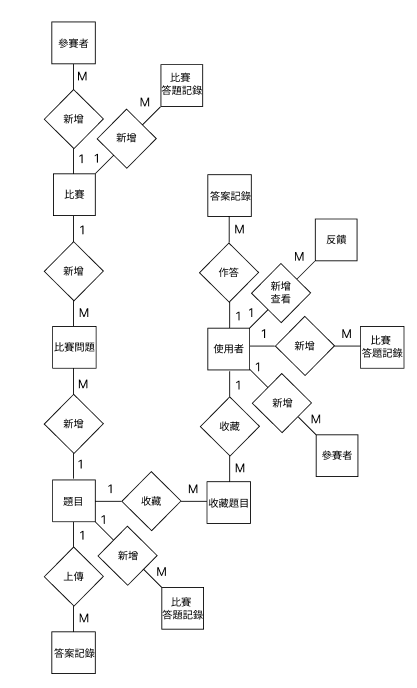
|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 6-1 | 功能名稱 | 反饋結果 |
| 功能說明 | 管理者將反饋結果發送給系統 | | |
| 輸入值 | 反饋結果 | | |
| 輸出值 | 反饋成功訊息 | | |

▼表 6-2-15 程序規格書-反饋內容

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 6-2 | 功能名稱 | 反饋內容 |
| 功能說明 | 系統將反饋結果的內容發送給使用者 | | |
| 輸入值 | 反饋內容 | | |
| 輸出值 | 反饋內容資料 | | |

第七章 資料模型

**7-1 實體關聯圖(Entity relationship diagram)**



* 圖 7-1-1 實體關聯圖

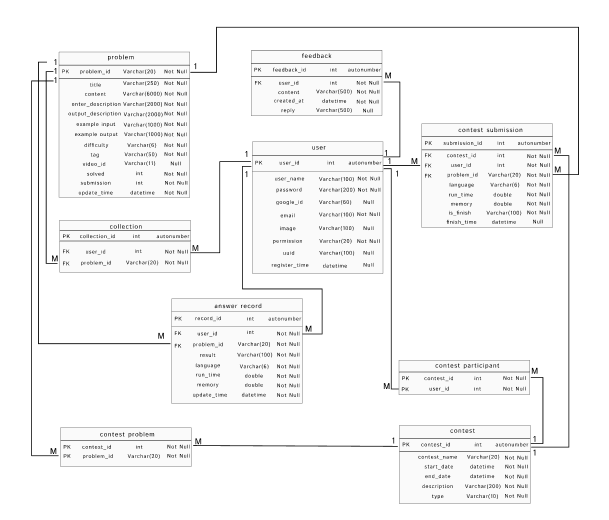
**7-2資料字典(Data dictionary)**

▼表 7-2-1 資料字典

|  |  |  |
| --- | --- | --- |
| **資料表編號** | **英文名稱** | **中文名稱** |
| **T01** | user | 使用者 |
| **T02** | feedback | 反饋 |
| **T03** | problem | 題目 |
| **T04** | answer record | 答題紀錄 |
| **T05** | collection | 收藏題目 |
| **T06** | contest problem | 比賽問題 |
| **T07** | contest | 比賽 |
| **T08** | contest participant | 參賽者 |
| **T09** | contest submission | 比賽答題記錄 |

第八章 資料庫設計

**8-1資料庫關聯表(Constraints)**



* 圖 8-1-1 資料庫關聯圖

**8-2表格及其Meta data**

▼表 8-2-1 user資料表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| T01 user | | | | | | |
| 欄位名稱 | 資料型態 | 長度 | 不可為空 | 主鍵 | 外鍵 | 備註 |
| user\_id | int |  | ü | ü |  |  |
| user\_name | varchar | 100 | ü |  |  |  |
| password | varchar | 200 | ü |  |  |  |
| google\_id | varchar | 60 |  |  |  |  |
| email | varchar | 100 | ü |  |  |  |
| image | varchar | 100 |  |  |  |  |
| permission | varchar | 20 | ü |  |  |  |
| uuid | varchar | 100 |  |  |  |  |
| register\_time | datetime |  |  |  |  |  |

▼表 8-2-2 feedback資料表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| T02 feedback | | | | | | |
| 欄位名稱 | 資料型態 | 長度 | 不可為空 | 主鍵 | 外鍵 | 備註 |
| feedback\_id | int |  | ü | ü |  |  |
| user\_id | int |  | ü |  | ü |  |
| content | varchar | 500 | ü |  |  |  |
| created\_at | datetime |  | ü |  |  |  |
| reply | varchar | 500 |  |  |  |  |

▼表 8-2-3 problem資料表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| T03 problem | | | | | | |
| 欄位名稱 | 資料型態 | 長度 | 不可為空 | 主鍵 | 外鍵 | 備註 |
| problem\_id | varchar | 20 | ü | ü |  |  |
| title | varchar | 250 | ü |  |  |  |
| content | varchar | 6000 | ü |  |  |  |
| enter  \_description | varchar | 2000 | ü |  |  |  |
| output  \_description | varchar | 2000 | ü |  |  |  |
| example  input | varchar | 1000 | ü |  |  |  |
| example  output | varchar | 1000 | ü |  |  |  |
| difficulty | varchar | 6 | ü |  |  |  |
| tag | varchar | 50 | ü |  |  |  |
| video\_id | varchar | 11 |  |  |  |  |
| solved | int |  | ü |  |  |  |
| submission | int |  | ü |  |  |  |
| update\_time | datetime |  | ü |  |  |  |

▼表 8-2-4 answer record資料表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| T04 answer record | | | | | | |
| 欄位名稱 | 資料型態 | 長度 | 不可為空 | 主鍵 | 外鍵 | 備註 |
| record\_id | int |  | ü | ü |  |  |
| user\_id | int |  | ü |  | ü |  |
| problem\_id | varchar | 20 | ü |  | ü |  |
| result | varchar | 100 | ü |  |  |  |
| language | varchar | 6 | ü |  |  |  |
| run\_time | double |  | ü |  |  |  |
| memory | double |  | ü |  |  |  |
| update\_time | datetime |  | ü |  |  |  |

▼表 8-2-5 collection資料表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| T05 collection | | | | | | |
| 欄位名稱 | 資料型態 | 長度 | 不可為空 | 主鍵 | 外鍵 | 備註 |
| collection \_id | int |  | ü | ü |  |  |
| user\_id | int |  | ü |  | ü |  |
| problem\_id | varchar | 20 | ü |  | ü |  |

▼表 8-2-6 contest problem資料表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| T06 contest problem | | | | | | |
| 欄位名稱 | 資料型態 | 長度 | 不可為空 | 主鍵 | 外鍵 | 備註 |
| contest\_id | int |  | ü | ü |  |  |
| problem\_id | varchar | 20 | ü | ü |  |  |

▼表 8-2-7 contest資料表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| T07 contest | | | | | | |
| 欄位名稱 | 資料型態 | 長度 | 不可為空 | 主鍵 | 外鍵 | 備註 |
| contest\_id | int |  | ü | ü |  |  |
| contest\_name | varchar | 20 | ü |  |  |  |
| start\_date | datetime |  | ü |  |  |  |
| end\_date | datetime |  | ü |  |  |  |
| description | varchar | 200 | ü |  |  |  |
| type | varchar | 10 | ü |  |  |  |

▼表 8-2-8 contest participant 資料表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| T08 contest participant | | | | | | |
| 欄位名稱 | 資料型態 | 長度 | 不可為空 | 主鍵 | 外鍵 | 備註 |
| contest\_id | int |  | ü | ü |  |  |
| user\_id | int |  | ü | ü |  |  |

▼表 8-2-9 contest submission 資料表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| T09 contest submission | | | | | | |
| 欄位名稱 | 資料型態 | 長度 | 不可為空 | 主鍵 | 外鍵 | 備註 |
| submission\_id | int |  | ü | ü |  |  |
| contest\_id | int |  | ü |  | ü |  |
| user\_id | int |  | ü |  | ü |  |
| problem\_id | varchar | 20 | ü |  | ü |  |
| language | varchar | 6 | ü |  |  |  |
| run\_time | double |  | ü |  |  |  |
| memory | double |  | ü |  |  |  |
| is\_finish | varchar | 100 | ü |  |  |  |
| finish\_time | datetime |  |  |  |  |  |

第九章 程式

**9-1 軟體架構與程式清單**

▼表 9-1-1 程式清單

|  |  |  |
| --- | --- | --- |
| 編號 | 檔案名稱 | 功能名稱 |
| 1 | app.py | 系統主要功能 |
| 2 | templates\problem\_list.html | 首頁 |
| 3 | templates\problem.html | 題目作答頁面 |
| 4 | templates\rank.html | 排行榜頁面 |
| 5 | services\auth\app.py | 使用者認證相關功能 |
| 6 | templates\login.html | 查詢電子郵件是否已註冊 |
| 7 | templates\login\_password.html | 登入密碼頁面 |
| 8 | templates\forget\_password.html | 忘記密碼頁面 |
| 9 | templates\verify\_forget\_password.html | 驗證忘記密碼郵件頁面 |
| 10 | templates\change\_password.html | 重設密碼頁面 |
| 11 |  |  |
|  | services\feedback\app.py | 回饋相關功能 |
|  | templates\feedback\_create\_form.html | 用戶回饋表單 |
|  | templates\admin\_dashboard.html | 管理回饋表單 |
|  | templates\reply\_feedback\_form.html | 回復回饋表單 |
|  | services\user\app.py | 用戶個人資料相關功能 |
|  | templates\user\_data.html | 用戶個人資料頁面 |
|  | services\manager\app.py | 管理者相關功能 |
|  | templates\add\_problem.html | 新增題目頁面 |
|  | templates\manager\_problem.html | 管理題目頁面 |
|  | templates\manager\_user.html | 管理使用者頁面 |
|  | crawler\add\_problem.py | 管理員上傳新的題目 |
|  | crawler\submit.py | 上傳題目到Judge平台 |
|  | crawler\get\_problem.py | 取得題目內容 |
|  | crawler\registration.py | 自動註冊機器人帳號 |
|  |  |  |
|  |  |  |

**9-2程式規格描述**

▼表 9-2-1 程式清單

|  |  |
| --- | --- |
| 路徑 | app.py |
| 用途 | 系統主要功能 |
| 部分程式碼 | |
| #主畫面  @app.route('/', methods=['GET'])  def index():  # 取得使用者的篩選條件  state = request.args.get('state', '\*', type=str)  onlinejudge = request.args.get('onlinejudge', '\*', type=str)  difficulty = request.args.get('difficulty', '\*', type=str)  search = request.args.get('search', '', type=str)  try:  # 檢查使用者有沒有登入  user\_id=session['User\_id']  sql\_problem\_command = """SELECT p.\*, acceptance\_data.acceptance\_rate, state.result  FROM `113-CodeAlchemy`.problem AS p"""  sql\_command=f""" LEFT JOIN (  SELECT problem\_id, user\_id, CASE  WHEN MAX(CASE WHEN result = 'Accepted' THEN 1 ELSE 0 END) = 1  THEN 'Accepted'  ELSE MAX(result) -- 如果沒有 'Accepted'，返回其他最高的 result  END AS result  FROM `113-CodeAlchemy`.`answer record` WHERE user\_id = '{user\_id}'  GROUP BY problem\_id, user\_id"""  sql\_problem\_command+=sql\_command  sql\_problem\_command+=""")AS state ON p.problem\_id = state.problem\_id"""  except:  sql\_problem\_command = """SELECT p.\*, acceptance\_data.acceptance\_rate  FROM `113-CodeAlchemy`.problem AS p"""  sql\_problem\_command += """  LEFT JOIN (  SELECT ar.problem\_id,  CONCAT(ROUND(SUM(CASE WHEN ar.result = 'Accepted' THEN 1 ELSE 0 END) / COUNT(ar.result) \* 100, 2), '%') AS acceptance\_rate  FROM `113-CodeAlchemy`.`answer record` AS ar  GROUP BY ar.problem\_id  ) AS acceptance\_data ON p.problem\_id = acceptance\_data.problem\_id"""  condition = ' WHERE ' # where前後的空格勿動  if search != '':  condition += f'p.title LIKE "%{search}%" AND '  if onlinejudge != '\*':  condition += f'SUBSTRING\_INDEX(p.problem\_id, "-", 1) = "{onlinejudge}" AND '  if difficulty != '\*':  condition += f'p.difficulty = "{difficulty}" AND '  try:  user\_id=session['User\_id']  if state == 'accept':  condition += f'state.result = "Accepted" AND '  elif state == 'error':  condition += f'state.result IS NOT NULL AND state.result != "Accepted" AND '  elif state == 'none':  condition += f'state.result IS NULL AND '  except:  pass  if condition == ' WHERE ':  condition = ''  else:  condition = condition[:condition.rfind(' AND ')]  sql\_problem\_command = sql\_problem\_command + condition  print(sql\_problem\_command)  data=db.get\_data(sql\_problem\_command)  # print(data)  # 預設第一頁  page = request.args.get('page', 1, type=int)  # 每頁顯示15列  per\_page = 15  start\_page = max(1, page - 1)  end\_page = min(page+3,math.ceil(paginate(data,page, per\_page)[1]/per\_page)+1)  paginated\_data = paginate(data,page, per\_page)[0]  #渲染網頁  return render\_template('problem\_list.html',data=paginated\_data,page=page,start\_page=start\_page,end\_page=end\_page,state=state,onlinejudge=onlinejudge,difficulty=difficulty,search=search)  '''run ok  #題目  @app.route('/problem',methods=['GET','POST'])  def problem():  if request.method=="POST":  # 從傳入封包取得資料  data = request.form  type = data.get('type')  problem\_id = data.get('problem\_id')  language = data.get('language')  code = data.get('code')  # 定義語言對應的文件擴展名字典  file\_extensions = {  'python': '.py',  'text/x-java': '.java',  'text/x-csrc': '.c',  'text/x-c++src': '.cpp'  }  # 生成 6 位數的亂碼  random\_code = str(uuid.uuid4())[:6]  # 生成隨機字串，第一個字母為英文字母  # 如果第一個字元不是英文字母，則重新生成，直到第一個字元為英文字母  while not random\_code[0].isalpha():  random\_code = str(uuid.uuid4())[:6]  # 構建文件路徑  file\_name = f'{random\_code}\_{problem\_id}{file\_extensions[language]}'  file\_path = os.path.join('./source', file\_name)  # 確保目錄存在  os.makedirs(os.path.dirname(file\_path), exist\_ok=True)  # 寫入內容到文件中  with open(file\_path, 'w') as file:  file.write(code)  if "ZJ" in file\_name:  # 調用 ZeroJudge\_submit 函數進行題目提交  score = ZeroJudge\_submit(file\_name,session['User\_id'])  if score and score[2]=='Accepted':  print()  elif "TIOJ" in file\_name:  run\_time=0  memory=0  score=TIOJ\_submit(file\_name,str(session['User\_id']))  if score and score[2]=='Accepted':  result=True  message="測試成功"  else :  result=False  message="測試失敗"  run\_time=score[3]  memory=score[4]  ensue=score[2]  if type == 'upload':  # 執行 SQL 插入語句  db.edit\_data(f'''  #INSERT INTO `answer record` (user\_id, problem\_id, result, language,run\_time,memory, update\_time)  #VALUES ('{session['User\_id']}','{problem\_id}','{ensue}', '{language}','{run\_time}','{memory}','{score[-1]}')  #''')    #return jsonify({'result':result,  # 'message':message,  # 'run\_time':run\_time,  # "memory":memory})  #else:  # problem\_id = request.args.get('problem\_id',type=str)  # sql\_problem\_command=f"SELECT \* FROM problem where problem\_id='{problem\_id}'"  # problem\_data=db.get\_data(sql\_problem\_command)  # example\_inputs = problem\_data[0][5].split('|||')  # example\_outputs = problem\_data[0][6].split('|||')  # video\_id = problem\_data[0][9]  # like = db.get\_data(f"SELECT IFNULL(COUNT(\*),0) FROM collection where problem\_id='{problem\_id}'")[0][0]  # return render\_template('./problem.html',data=problem\_data,example\_inputs=example\_inputs,example\_outputs=example\_outputs,like=like,video\_id=video\_id)  '''  ####################  '''  # 確認使用者有登入  @app.route('/get\_user\_id', methods=['GET'])  def get\_user\_id():  if 'User\_id' in session:  return jsonify({'user\_id': session['User\_id']})  else:  return jsonify({'error': 'User not logged in'}), 401    # 題目  @app.route('/problem', methods=['GET', 'POST'])  def problem():  if request.method == "POST":  # 從傳入封包取得資料  data = request.form  problem\_id = data.get('problem\_id')  language = data.get('language')  code = data.get('code')  source = data.get('source')  contest\_id = data.get('contest\_id')  ###########################  # 獲取完整的 URL  # full\_url = request.form.get('fullUrl') # 使用 request.form 獲取資料  # print(f"POST request - Full URL: {full\_url}") # 僅打印一次  # # 解析 URL 以獲取查詢參數  # parsed\_url = urlparse(full\_url)  # query\_params = parse\_qs(parsed\_url.query)  # # 獲取 source 和 contest\_id  # source = query\_params.get('source', [None])[0] # 預設為 None  # contest\_id = query\_params.get('contest\_id', [None])[0] # 預設為 None  # 確認是否收到正確的參數  print(f"Source: {source}, Contest ID: {contest\_id}") # 確認 source 和 contest\_id  ################################  # 定義語言對應的文件擴展名字典  file\_extensions = {  'python': '.py',  'text/x-java': '.java',  'text/x-csrc': '.c',  'text/x-c++src': '.cpp'  }  # 生成 6 位數的亂碼  random\_code = str(uuid.uuid4())[:6]  while not random\_code[0].isalpha():  random\_code = str(uuid.uuid4())[:6]  # 構建文件路徑  file\_name = f'{random\_code}\_{problem\_id}{file\_extensions[language]}'  file\_path = os.path.join('./source', file\_name)  # 確保目錄存在  os.makedirs(os.path.dirname(file\_path), exist\_ok=True)  # 寫入內容到文件中  with open(file\_path, 'w') as file:  file.write(code)    if "ZJ" in file\_name:  score = ZeroJudge\_submit(file\_name, session['User\_id'])  elif "TIOJ" in file\_name:  score = TIOJ\_submit(file\_name, str(session['User\_id']))  result = False  message = "測試失敗"  run\_time = 0  memory = 0  ensue = "Rejected"  if score and score[2] == 'Accepted':  result = True  message = "測試成功"  run\_time = score[3]  memory = score[4]  ensue = score[2]  # 確認提交來源並插入到正確的資料表  if source == 'contest':  print(f"決策時的 Source 值: {source}") # 確認 source 的值  # 插入記錄到 contest\_submission 資料表  db.edit\_data(f'''  INSERT INTO `contest submission` (contest\_id, user\_id, problem\_id, language, run\_time, memory, is\_finish, finish\_time)  VALUES ('{contest\_id}', '{session['User\_id']}', '{problem\_id}', '{language}', '{run\_time}', '{memory}', '{ensue}', '{score[-1]}')  ''')  print("成功插入到 contest\_submission 資料表") # 用於確認的訊息  else:  print(f"決策時的 Source 值: {source}") # 確認 source 的值  # 插入記錄到 answer record 資料表  db.edit\_data(f'''  INSERT INTO `answer record` (user\_id, problem\_id, result, language, run\_time, memory, update\_time)  VALUES ('{session['User\_id']}', '{problem\_id}', '{ensue}', '{language}', '{run\_time}', '{memory}', '{score[-1]}')  ''')  print("成功插入到 answer record 資料表") # 用於確認的訊息  return jsonify({  'result': result,  'message': message,  'run\_time': run\_time,  'memory': memory  })  else:  # GET 請求處理邏輯    problem\_id = request.args.get('problem\_id', type=str)  source = request.args.get('source')  contest\_id = request.args.get('contest\_id')  sql\_problem\_command = f"SELECT \* FROM problem WHERE problem\_id='{problem\_id}'"  problem\_data = db.get\_data(sql\_problem\_command)  example\_inputs = problem\_data[0][5].split('|||')  example\_outputs = problem\_data[0][6].split('|||')  video\_id = problem\_data[0][9]  like = db.get\_data(f"SELECT IFNULL(COUNT(\*), 0) FROM collection WHERE problem\_id='{problem\_id}'")[0][0]  return render\_template('./problem.html', data=problem\_data, example\_inputs=example\_inputs, example\_outputs=example\_outputs, like=like, video\_id=video\_id)  ###################  @app.route('/answer\_record',methods=['GET'])  def answer\_record():  problem\_id = request.args.get('problem\_id',type=str)  sql\_problem\_command=f"""SELECT record\_id,u.user\_id,image,user\_name,result,language,run\_time,memory,ar.update\_time FROM `113-CodeAlchemy`.`answer record` as ar  left join `user` as u  on ar.user\_id=u.user\_id  where problem\_id='{problem\_id}'  order by update\_time desc"""  data=db.get\_data(sql\_problem\_command)  return render\_template('./answer\_record.html',data=data)    # 收藏  @app.route('/add\_to\_collection', methods=['POST'])  def add\_to\_collection():  data = request.get\_json()  problem\_id = data.get('problem\_id')  try:  user\_id = session['User\_id']  if db.get\_data(f"SELECT IFNULL(COUNT(\*),0) FROM collection where problem\_id='{problem\_id}' and user\_id='{user\_id}'")[0][0]==0:  sql\_command=f"INSERT INTO collection(user\_id,problem\_id) VALUES ('{user\_id}','{problem\_id}')"  else:  sql\_command=f"DELETE FROM collection WHERE problem\_id = '{problem\_id}' and user\_id='{user\_id}'"  db.edit\_data(sql\_command)  return jsonify({'message': 'Item added to collection'}), 201  except KeyError:  return jsonify({'error': 'Missing item\_id or user\_id'}), 400    @app.route('/dolos', methods=['GET'])  def problem\_dolos():  problem\_id=request.args.get('problem\_id',type=str)  zip=dolos.create\_zip(problem\_id)  url=dolos.submit\_to\_dolos(zip[0],zip[1])  return (redirect(url))  @app.route("/rank")  def rank():  data=db.get\_data(f'''  SELECT  u.user\_id,  u.user\_name,  u.image,  SUM(CASE WHEN ar.result = 'Accepted' THEN 1 ELSE 0 END) AS 正確答題數,  CONCAT(ROUND(SUM(CASE WHEN ar.result = 'Accepted' THEN 1 ELSE 0 END) / COUNT(\*) \* 100, 2), '%') AS 答題正確率,  round(avg(ar.run\_time),2) as 平均執行時間,  round(avg(ar.memory),2) as 平均使用記憶體  FROM  `113-CodeAlchemy`.`answer record` AS ar  LEFT JOIN  `user` AS u ON u.user\_id = ar.user\_id  GROUP BY  u.user\_id, u.user\_name  order by 正確答題數 desc;  ''')  return render\_template('./rank.html',data=data) | |

|  |  |
| --- | --- |
| 路徑 | templates\problem\_list.html |
| 用途 | 首頁前端 |
| 部分程式碼 | |
| <div class="container">  <div class="row mt-4">  <div class="col-3">  <select class="form-select" id="state" onchange="location.href=this.options[this.selectedIndex].value">  <option selected>狀態</option>  <option  value="?page={{ page }}&state=\*&onlinejudge={{ onlinejudge }}&difficulty={{ difficulty }}&search={{ search }}">  All  </option>  <option  value="?page={{ page }}&state=accept&onlinejudge={{ onlinejudge }}&difficulty={{ difficulty }}&search={{ search }}">  通過  </option>  <option  value="?page={{ page }}&state=error&onlinejudge={{ onlinejudge }}&difficulty={{ difficulty }}&search={{ search }}">  錯誤  </option>  <option  value="?page={{ page }}&state=none&onlinejudge={{ onlinejudge }}&difficulty={{ difficulty }}&search={{ search }}">  未做  </option>  </select>  </div>  <div class="col-3">  <select class="form-select" id="onlinejudge"  onchange="location.href=this.options[this.selectedIndex].value">  <option selected>OnlineJudge</option>  <option  value="?page=1&state={{ state }}&onlinejudge=\*&difficulty={{ difficulty }}&search={{ search }}">  All  </option>  <option  value="?page=1&state={{ state }}&onlinejudge=ZJ&difficulty={{ difficulty }}&search={{ search }}">  ZeroJudge  </option>  <option  value="?page=1&state={{ state }}&onlinejudge=TIOJ&difficulty={{ difficulty }}&search={{ search }}">  TIOJ  </option>  <option  value="?page=1&state={{ state }}&onlinejudge=LeetCode&difficulty={{ difficulty }}&search={{ search }}">  LeetCode  </option>  </select>  </div>  <div class="col-3">  <select class="form-select" id="difficulty"  onchange="location.href=this.options[this.selectedIndex].value">  <option selected>難度</option>  <option  value="?page={{ page }}&state={{ state }}&onlinejudge={{ onlinejudge }}&difficulty=\*&search={{ search }}">  All  </option>  <option  value="?page={{ page }}&state={{ state }}&onlinejudge={{ onlinejudge }}&difficulty=Hard&search={{ search }}">  Hard  </option>  <option  value="?page={{ page }}&state={{ state }}&onlinejudge={{ onlinejudge }}&difficulty=Normal&search={{ search }}">  Middle  </option>  <option  value="?page={{ page }}&state={{ state }}&onlinejudge={{ onlinejudge }}&difficulty=Easy&search={{ search }}">  Easy  </option>  </select>  </div>  <div class="col-3">  <form class="d-flex" action="" method="get">  <input type="hidden" name="page" value="1">  <input type="hidden" name="state" value="{{ state }}">  <input type="hidden" name="onlinejudge" value="{{ onlinejudge }}">  <input type="hidden" name="difficulty" value="{{ difficulty }}">  <input class="form-control me-2" name="search" placeholder="Search" aria-label="Search" id="search" value="{{ search }}">  </form>  </div>  </div>  <!-- ------- -->  <!-- 題目列表 -->  <!-- ------- -->  <div class="table-responsive">  <table id="problem\_list" class="table table-hover justify-content-center mt-4">  <thead>  <tr>  <td>通過</td>  <td>OJ</td>  <td>編號</td>  <td>名稱</td>  <td>難度</td>  <td>標籤</td>  <td>通過率</td>  <td>上傳日期</td>  {% if (session['Permission']=='Manager') %}  <td>修改</td>  <td>刪除</td>  {% else %}  {% endif %}  </tr>  </thead>  <tbody>  {% for item in data %}  <tr onclick="window.location='./problem?problem\_id={{item[0]}}'">  <td class="text-success">  {% if item['passed'] %}  <svg xmlns="http://www.w3.org/2000/svg" width="16" height="16" fill="currentColor"  class="bi bi-check-lg" viewBox="0 0 16 16">  <path  d="M12.736 3.97a.733.733 0 0 1 1.047 0c.286.289.29.756.01 1.05L7.88 12.01a.733.733 0 0 1-1.065.02L3.217 8.384a.757.757 0 0 1 0-1.06.733.733 0 0 1 1.047 0l3.052 3.093 5.4-6.425z" />  </svg>  {% endif %}  </td>  <td>  {% if item[0].split('-')[0] == 'ZJ' %}  Zero<br>Judge  {% elif item[0].split('-')[0] == 'TIOJ' %}  TIOJ  {% endif %}  </td>  <td>{{ item[0] }}</td> <!--編號-->  <td>{{ item[1] }}</td> <!--名稱-->  <td>{{ item[7] }}</td> <!--難度-->  <td>{{ item[8] }}</td> <!--標籤-->  <td>  {{ item[12] }}<!--通過率-->  </td>  <td>{{ item[11].strftime('%Y-%m-%d') }}</td>  {% if (session['Permission']=='Manager') %}  <!-- 修改 -->  <td><a class="btn btn-success" href="/manager/problem?problem\_id={{item[0]}}" role="button"><svg  xmlns="http://www.w3.org/2000/svg" width="16" height="16" fill="currentColor"  class="bi bi-pencil" viewBox="0 0 16 16">  <path  d="M12.146.146a.5.5 0 0 1 .708 0l3 3a.5.5 0 0 1 0 .708l-10 10a.5.5 0 0 1-.168.11l-5 2a.5.5 0 0 1-.65-.65l2-5a.5.5 0 0 1 .11-.168zM11.207 2.5 13.5 4.793 14.793 3.5 12.5 1.207zm1.586 3L10.5 3.207 4 9.707V10h.5a.5.5 0 0 1 .5.5v.5h.5a.5.5 0 0 1 .5.5v.5h.293zm-9.761 5.175-.106.106-1.528 3.821 3.821-1.528.106-.106A.5.5 0 0 1 5 12.5V12h-.5a.5.5 0 0 1-.5-.5V11h-.5a.5.5 0 0 1-.468-.325" />  </svg></a></td>  <!-- 刪除 -->  <td><a class="btn btn-danger" href="/manager/delete?problem\_id={{item[0]}}" role="button"><svg xmlns="http://www.w3.org/2000/svg"  width="16" height="16" fill="currentColor" class="bi bi-trash" viewBox="0 0 16 16">  <path  d="M5.5 5.5A.5.5 0 0 1 6 6v6a.5.5 0 0 1-1 0V6a.5.5 0 0 1 .5-.5m2.5 0a.5.5 0 0 1 .5.5v6a.5.5 0 0 1-1 0V6a.5.5 0 0 1 .5-.5m3 .5a.5.5 0 0 0-1 0v6a.5.5 0 0 0 1 0z" />  <path  d="M14.5 3a1 1 0 0 1-1 1H13v9a2 2 0 0 1-2 2H5a2 2 0 0 1-2-2V4h-.5a1 1 0 0 1-1-1V2a1 1 0 0 1 1-1H6a1 1 0 0 1 1-1h2a1 1 0 0 1 1 1h3.5a1 1 0 0 1 1 1zM4.118 4 4 4.059V13a1 1 0 0 0 1 1h6a1 1 0 0 0 1-1V4.059L11.882 4zM2.5 3h11V2h-11z" />  </svg></a></td>  {% else %}  {% endif %}  </tr>  {% endfor %}  </tbody>  </table>  </div>  <!-- 題目列表分頁 -->  <div class="d-flex justify-content-center">  <nav aria-label="Page navigation">  <ul class="pagination">  {% if page > 1 %}  <li class="page-item"><a class="page-link"  href="?page=1&state={{state}}&onlinejudge={{onlinejudge}}&difficulty={{difficulty}}&search={{search}}">第一頁</a>  </li>  {% else %}  <li class="page-item disabled"><span class="page-link">第一頁</span></li>  {% endif %}  {% for p in range(start\_page, end\_page) %}  {% if p == page %}  <li class="page-item active" aria-current="page"><span class="page-link">{{ p }}</span></li>  {% else %}  <li class="page-item"><a class="page-link"  href="?page={{p}}&state={{state}}&onlinejudge={{onlinejudge}}&difficulty={{difficulty}}&search={{search}}">{{  p }}</a></li>  {% endif %}  {% endfor %}  {% if page < end\_page-1 %} <li class="page-item disabled"><a class="page-link" href="#">...</a></li>  <li class="page-item"><a class="page-link"  href="?page={{page+1}}&state={{state}}&onlinejudge={{onlinejudge}}&difficulty={{difficulty}}&search={{search}}">下一頁</a>  </li>  {%else%}  <li class="page-item disabled"><span class="page-link">下一頁</span></li>  {%endif%}  </ul>  </nav>  </div>  </div> | |

|  |  |
| --- | --- |
| 路徑 | templates\problem.html |
| 用途 | 題目作答頁面前端 |
| 部分程式碼 | |
| <div class="container-md">          <div class="row">              <div class="col">                  <div class="row">                      <div class="col-6">                          <div id="problem\_id" style="display: none;">{{data[0][0]}}</div>                          <h1 class="display-6 mt-4">{{data[0][0]}}<br>{{data[0][1]}}</h1>                      </div>                      <!-- 收藏按鈕 -->                      <div class="col-6 d-flex justify-content-center align-items-center">                          <button class="Btn">                              <span class="leftContainer">                                  <svg fill="white" viewBox="0 0 512 512" height="1em" xmlns="http://www.w3.org/2000/svg">                                      <path                                          d="M47.6 300.4L228.3 469.1c7.5 7 17.4 10.9 27.7 10.9s20.2-3.9 27.7-10.9L464.4 300.4c30.4-28.3 47.6-68 47.6-109.5v-5.8c0-69.9-50.5-129.5-119.4-141C347 36.5 300.6 51.4 268 84L256 96 244 84c-32.6-32.6-79-47.5-124.6-39.9C50.5 55.6 0 115.2 0 185.1v5.8c0 41.5 17.2 81.2 47.6 109.5z">                                      </path>                                  </svg>                                  <span class="like">Like</span>                              </span>                              <span class="likeCount">                                  {{like}}                              </span>                          </button>                      </div>                      <!-- 分頁 -->                      <div class="nav nav-tabs nav-justified">                          <button class="nav-link active" data-bs-toggle="tab" data-bs-target="#menu1"                              aria-selected="true">題目</button>                          <button class="nav-link" data-bs-toggle="tab" data-bs-target="#menu2"                              aria-selected="false">解題影片</button>                      </div>                      <div class="tab-content">                          <!-- 題目 -->                          <div class="tab-pane fade show active" id="menu1">                              <div class="col-12">                                  <h3 class="mt-4">題目說明:<br></h3>                                  <h4 class="mt-4 overflow-auto">{{data.0.2|safe}}</h4>                                  <h3 class="mt-4">輸入說明:<br></h3>                                  <h4 class="mt-4 overflow-auto">{{data.0.3|safe}}</h4>                                  <h3 class="mt-4">輸出說明:<br></h3>                                  <h4 class="mt-4 overflow-auto">{{data.0.4|safe}}</h4>                              </div>                              {% for n in range(example\_inputs|length) %}                              <h5 class="mt-4 overflow-auto">範例{{n+1}}</h5>                              <div class="border-start border-4">                                  <h6 class="ms-2 mt-4 overflow-auto">輸入</h6>                                  <h6 class="ms-2 mt-4 overflow-auto code-font">{{ example\_inputs[n]|safe }}</h6>                                  <h5 class="ms-2 mt-4 overflow-auto">輸出</h5>                                  <h6 class="ms-2 mt-4 overflow-auto code-font">{{ example\_outputs[n]|safe }}</h6>                              </div>                              {%endfor%}                          </div>                          <!-- 教學影片 -->                          <div class="tab-pane fade" id="menu2">                              <div class="col-12">                                  <iframe width="560" height="315"                                      src="https://www.youtube.com/embed/nZqMjBUp3Fs?si=\_5ERs1-g2\_hBQgmT"                                      title="YouTube video player" frameborder="0"                                      allow="accelerometer; autoplay; clipboard-write; encrypted-media; gyroscope; picture-in-picture; web-share"                                      referrerpolicy="strict-origin-when-cross-origin" allowfullscreen></iframe>                              </div>                          </div>                      </div>                  </div>              </div>              <div class="col mt-5">                  <div class="row d-flex justify-content-center">                      <div class="btn-group" role="group" aria-label="Basic example">                          <div class="btn-group">                              <button type="button" class="btn btn-primary dropdown-toggle" data-bs-toggle="dropdown"                                  aria-expanded="false">                                  <svg xmlns="http://www.w3.org/2000/svg" width="16" height="16" fill="currentColor"                                      class="bi bi-code-square" viewBox="0 0 16 16">                                      <path                                          d="M14 1a1 1 0 0 1 1 1v12a1 1 0 0 1-1 1H2a1 1 0 0 1-1-1V2a1 1 0 0 1 1-1zM2 0a2 2 0 0 0-2 2v12a2 2 0 0 0 2 2h12a2 2 0 0 0 2-2V2a2 2 0 0 0-2-2z" />                                      <path                                          d="M6.854 4.646a.5.5 0 0 1 0 .708L4.207 8l2.647 2.646a.5.5 0 0 1-.708.708l-3-3a.5.5 0 0 1 0-.708l3-3a.5.5 0 0 1 .708 0m2.292 0a.5.5 0 0 0 0 .708L11.793 8l-2.647 2.646a.5.5 0 0 0 .708.708l3-3a.5.5 0 0 1 0-.708l-3-3a.5.5 0 0 0-.708 0" />                                  </svg>                                  語言                              </button>                              <ul class="dropdown-menu" id="select\_language\_button">                                  <li><a class="dropdown-item" data-mode="python" value="python">Python</a></li>                                  <li><a class="dropdown-item" data-mode="text/x-csrc" value="c">C</a></li>                                  <li><a class="dropdown-item" data-mode="text/x-c++src" value="cpp">C++</a></li>                                  <li><a class="dropdown-item" data-mode="text/x-java" value="java">Java</a></li>                              </ul>                          </div>                          <button type="button" class="btn btn-outline-primary" id="test\_btn" data-bs-toggle="modal"                              data-bs-target="#staticBackdrop">                              <svg xmlns="http://www.w3.org/2000/svg" width="16" height="16" fill="currentColor"                                  class="bi bi-code-slash" viewBox="0 0 16 16">                                  <path                                      d="M10.478 1.647a.5.5 0 1 0-.956-.294l-4 13a.5.5 0 0 0 .956.294zM4.854 4.146a.5.5 0 0 1 0 .708L1.707 8l3.147 3.146a.5.5 0 0 1-.708.708l-3.5-3.5a.5.5 0 0 1 0-.708l3.5-3.5a.5.5 0 0 1 .708 0m6.292 0a.5.5 0 0 0 0 .708L14.293 8l-3.147 3.146a.5.5 0 0 0 .708.708l3.5-3.5a.5.5 0 0 0 0-.708l-3.5-3.5a.5.5 0 0 0-.708 0" />                              </svg>測試</button>                          <!-- Modal -->                          <!-- 進度條視窗 -->                          <div class="modal fade" id="staticBackdrop" data-bs-backdrop="static" data-bs-keyboard="false"                              tabindex="-1" aria-labelledby="staticBackdropLabel" aria-hidden="true">                              <div class="modal-dialog">                                  <div class="modal-content">                                      <div class="modal-header">                                          <h5 class="modal-title" id="staticBackdropLabel">測試中</h5>                                      </div>                                      <div class="modal-body">                                          <div aria-label="Orange and tan hamster running in a metal wheel" role="img"                                              class="wheel-and-hamster">                                              <div class="wheel"></div>                                              <div class="hamster">                                                  <div class="hamster\_\_body">                                                      <div class="hamster\_\_head">                                                          <div class="hamster\_\_ear"></div>                                                          <div class="hamster\_\_eye"></div>                                                          <div class="hamster\_\_nose"></div>                                                      </div>                                                      <div class="hamster\_\_limb hamster\_\_limb--fr"></div>                                                      <div class="hamster\_\_limb hamster\_\_limb--fl"></div>                                                      <div class="hamster\_\_limb hamster\_\_limb--br"></div>                                                      <div class="hamster\_\_limb hamster\_\_limb--bl"></div>                                                      <div class="hamster\_\_tail"></div>                                                  </div>                                              </div>                                              <div class="spoke"></div>                                          </div>                                      </div>                                      <div class="modal-footer">                                          <button type="button" class="btn btn-secondary" data-bs-dismiss="modal"                                              id="stop\_btn">停止</button>                                      </div>                                  </div>                              </div>                          </div>                          <button type="button" class="btn btn-outline-success" id="upload\_btn" data-bs-toggle="modal"                              data-bs-target="#staticBackdrop"><svg xmlns="http://www.w3.org/2000/svg" width="16"                                  height="16" fill="currentColor" class="bi bi-cloud-arrow-up" viewBox="0 0 16 16">                                  <path fill-rule="evenodd"                                      d="M7.646 5.146a.5.5 0 0 1 .708 0l2 2a.5.5 0 0 1-.708.708L8.5 6.707V10.5a.5.5 0 0 1-1 0V6.707L6.354 7.854a.5.5 0 1 1-.708-.708z" />                                  <path                                      d="M4.406 3.342A5.53 5.53 0 0 1 8 2c2.69 0 4.923 2 5.166 4.579C14.758 6.804 16 8.137 16 9.773 16 11.569 14.502 13 12.687 13H3.781C1.708 13 0 11.366 0 9.318c0-1.763 1.266-3.223 2.942-3.593.143-.863.698-1.723 1.464-2.383m.653.757c-.757.653-1.153 1.44-1.153 2.056v.448l-.445.049C2.064 6.805 1 7.952 1 9.318 1 10.785 2.23 12 3.781 12h8.906C13.98 12 15 10.988 15 9.773c0-1.216-1.02-2.228-2.313-2.228h-.5v-.5C12.188 4.825 10.328 3 8 3a4.53 4.53 0 0 0-2.941 1.1z" />                              </svg>上傳</button>                      </div>                      <p></p>                      <a href="/dolos?problem\_id={{data[0][0]}}" class="btn btn-primary w-70" role="button"                          aria-disabled="true">程式碼分析</a>                      <p></p>                      <a href='./answer\_record?problem\_id={{data[0][0]}}' class="btn btn-primary w-70" role="button"                          aria-disabled="true">作答情況</a>                      <!-- 程式輸入框 -->                      <div class="row border border-3 round rounded-2 mt-5">                          <textarea id="editor" name="editor">  </textarea>                      </div>                      <div id="result\_container">                          <div id="passed" class="mt-5 bg-success p-3 text-white rounded" style="display: none;">                              <h4 id="passed\_status"></h4>                              <p id="passed\_run\_time"></p>                              <p id="passed\_memory"></p>                          </div>                          <div id="failed" class="mt-5 bg-danger p-3 text-white rounded" style="display: none;">                              <h4 id="failed\_status"></h4>                              <p id="failed\_error\_reason"></p>                          </div>                      </div>                  </div>              </div>          </div>      </div> | |

|  |  |
| --- | --- |
| 路徑 | templates\rank.html |
| 用途 | 排行榜前端 |
| 部分程式碼 | |
| <div class="container">          <!-- ------- -->          <!-- 作答紀錄列表 -->          <!-- ------- -->          <div class="table-responsive">              <table id="problem\_list" class="table table-hover justify-content-center mt-4">                  <thead>                      <tr>                          <td>名次</td>                          <td>用戶名稱</td>                          <td>正確答題數</td>                          <td>正確率</td>                          <td>平均執行時間</td>                          <td>平均使用記憶體</td>                      </tr>                  </thead>                  <tbody>                      {% for item in data %}                      <tr>                          <td>{{ loop.index }}</td>                          <td>{% if item[2] is none %}                              <img src="./static/user\_icon/default.png" class="circle-img">                          {% else %}                              <img src="{{ item[2] }}" class="circle-img">                          {% endif %}                          {{ item[1] }}                          </td>                          <td>{{ item[3] }}</td>                          <td>{{ item[4] }}</td>                          <td>{{ item[5] }} ms</td>                          <td>{{ item[6] }} MB</td>                      </tr>                      {% endfor %}                  </tbody>              </table>          </div>      </div> | |

|  |  |
| --- | --- |
| 路徑 | services\auth\app.py |
| 用途 | 使用者認證相關 |
| 部分程式碼 | |
| # google登入  oauth = OAuth(app)  google = oauth.register(  name='google',  client\_id=GOOGLE\_CELENT\_ID,  client\_secret=GOOGLE\_CELENT\_SERRET,  client\_kwargs= {"scope": "openid email profile"},  server\_metadata\_url= 'https://accounts.google.com/.well-known/openid-configuration')  # 查詢電子郵件有沒有註冊過  @auth\_bp.route('/login', methods=['GET', 'POST'])  def login():  # 如果是 POST 請求，表示用戶提交了表單  if request.method == "POST":  # 從表單中獲取電子郵件  Email = request.form['Email']  # 構建 SQL 查詢命令，查詢使用者資料庫中是否存在這個電子郵件  sql\_user\_command = f"SELECT \* FROM user where email='{Email}'"  # 執行 SQL 查詢，獲取使用者資料  user\_data = db.get\_data(sql\_user\_command)  # 如果查詢結果有一條記錄，表示用戶已經註冊過  if len(user\_data) == 1:  # 將電子郵件存入 session  session['Email'] = Email  # 重定向到輸入密碼頁面  return redirect('/auth/login\_password')  # 如果用戶尚未註冊，將用戶重定向到註冊頁面  else:  return redirect('/auth/register')  else:  # 如果是 GET 請求，即用戶訪問該路由時，返回登入表單模板  return render\_template('login.html')  # Google 登入路由  @auth\_bp.route('/google')  def login\_google():  # 設置重定向 URI，用於 Google 登入完成後的跳轉  redirect\_uri = url\_for('auth.authorize', \_external=True)  # 呼叫 OAuth 库的 authorize\_redirect() 方法，將用戶導向 Google 登入頁面  return google.authorize\_redirect(redirect\_uri)  # Google 登入回調路由  @auth\_bp.route('/google-callback')  def authorize():  # 取得 Google 登入的存取令牌  token = google.authorize\_access\_token()  # 通過存取令牌向 Google API 發送請求，獲取用戶資訊  user\_info = google.get('https://www.googleapis.com/oauth2/v1/userinfo').json()  # 從用戶資訊中提取必要的數據  Goole\_ID = user\_info['id']  Email = user\_info['email']  imgs = user\_info['picture']  # 在數據庫中查詢是否有使用該郵箱註冊過  sql\_user\_command = f"SELECT \* FROM user where email='{Email}'"  user\_data = db.get\_data(sql\_user\_command)  # 如果郵箱已註冊  if len(user\_data) == 1:  # 將用戶資訊存入 session  session['User\_id']=user\_data[0][0]  session['Email'] = Email  session['logged\_in'] = True  session['User\_name'] = user\_data[0][1]  session['Permission']=user\_data[0][6]  session['google\_id'] = Goole\_ID  session['imgs'] = imgs  # 如果數據庫中的 Google ID 與 Google 返回的 ID 一致，則重定向到首頁  if user\_data[0][3] == Goole\_ID:  return redirect('/')  # 否則，重定向到連結 Google 賬戶頁面  else:  return redirect('/auth/connect\_google')  # 如果郵箱尚未註冊，則重定向到註冊頁面  else:  return redirect('/auth/register')  # 連結 Google 賬戶路由  @auth\_bp.route('/connect\_google',methods=['GET','POST'])  def connect\_google():  # 從 session 中獲取 Google ID、郵箱和用戶圖片  google\_id = session.get('google\_id')  Email = session.get('Email')  imgs = session.get('imgs')  # 如果是 POST 請求  if request.method=="POST":  # 獲取輸入的密碼  Password=request.form['Password']  # 查詢數據庫，確認密碼是否正確  sql\_common = f"SELECT \* FROM user where email='{Email}'"  user\_data = db.get\_data(sql\_common)  # 如果密碼驗證成功  if check\_password\_hash(db.get\_data(sql\_common)[0][2],Password):  # 將用戶標記為已登入  session['logged\_in'] = True  # 將用戶名稱存入 session  session['User\_name'] = user\_data[0][1]  # 將用戶 ID 存入 session  session['User\_id'] = user\_data[0][0]  session['Permission']=user\_data[0][6]  # 更新數據庫中的 Google ID 和圖片  sql\_user\_command = f"UPDATE user SET google\_id = '{google\_id}', image = '{imgs}' WHERE email = '{Email}'"  db.edit\_data(sql\_user\_command)  # 重定向到首頁  return redirect('/')  # 如果密碼驗證失敗  else:  result = '密碼錯誤'  return render\_template('connect\_google.html',result=result)  # 如果是 GET 請求，直接渲染連結 Google 賬戶頁面  return render\_template('connect\_google.html')  # 登入密碼頁面路由  @auth\_bp.route('/login\_password',methods=['GET','POST'])  def login\_password():  # 如果是 POST 請求  if request.method=="POST":  # 從 session 中獲取郵箱  Email = session.get('Email')  # 獲取輸入的密碼  Password=request.form['Password']  # 檢查是否勾選了「記住我」  try:  Rememberme=request.form['Rememberme']  Rememberme=1  except Exception:  Rememberme=0  # 查詢使用者資料  sql\_common=f"SELECT \* FROM user where email='{Email}'"  user\_data=db.get\_data(sql\_common)  # 如果密碼驗證成功  if check\_password\_hash(db.get\_data(sql\_common)[0][2],Password):  # 將用戶標記為已登入  session['logged\_in']=True  # 將用戶名稱存入 session  session['User\_id']=user\_data[0][0]  session['User\_name']=user\_data[0][1]  # 將用戶 ID 存入 session  session['User\_id']=user\_data[0][0]  # 將用戶權限存入 session  session['Permission']=user\_data[0][6]  # 如果使用者有勾選「記住我」  if Rememberme==1:  resp = make\_response(redirect('/'))  # 設置 cookie 有效期為 30 天  resp.set\_cookie('logged\_in','True',max\_age=60\*60\*24\*30)  resp.set\_cookie('user\_name',user\_data[0][1],max\_age=60\*60\*24\*30)  resp.set\_cookie('user\_id',str(user\_data[0][0]),max\_age=60\*60\*24\*30)  return resp  else:  return redirect('/')  # 如果密碼驗證失敗  else:  result='密碼錯誤'  return render\_template('login\_password.html',result=result)  else:  return render\_template('login\_password.html')  # 註冊頁面路由  @auth\_bp.route('/register' ,methods=['GET','POST'])  def register():  if request.method == "POST":  user\_name=request.form['Username']  Email=request.form['Email']  Password=request.form['Password']  # 檢查郵箱是否已經註冊過  if db.get\_data(f"SELECT \* FROM user where email='{Email}'"):  result='此Email已經註冊過'  return redirect('/auth/login')  else:  token=str(uuid.uuid4())  # 在數據庫中插入新用戶資料  sql\_user\_command=f"INSERT INTO user(user\_name,password,email,permission,uuid) VALUES ('{user\_name}','{generate\_password\_hash(Password)}','{Email}','Default user','{token}')"  db.edit\_data(sql\_user\_command)  html=f'http://140.131.114.141/auth/verify\_register?uuid={token}'  msg\_title = 'Welcome to CodeAlchemy'  msg\_recipients=[Email]  msg\_html =f'<p>親愛的 {user\_name}，<br>感謝您註冊成為我們平台的一員！為了確保您的帳戶安全，請點擊以下連結驗證您的電子郵件地址：<a href="{html}">驗證連結</a>。<br>如果您無法點擊上述連結，請將以下網址複製並粘貼到瀏覽器地址欄中：<a href="{html}">{html}</a>。<br>請完成這一步驟以啟用您的帳戶。如果您遇到任何問題或需要協助，請隨時聯繫我們的客戶服務團隊，我們將竭誠為您服務。<br>謝謝您的合作！<br>祝您有個愉快的體驗！</p>'  msg = Message(  subject=msg\_title,  sender = 'codealchemyproject@gmail.com',  recipients=msg\_recipients,  html=msg\_html,  charset='UTF-8'  )  mail.send(msg)  result='註冊成功'  return render\_template('register\_result.html',result=result)  else:  return render\_template('register.html')  # 註冊驗證頁面路由  @auth\_bp.route('/verify\_register',methods=['GET'])  def verify\_register():  # 獲得 uuid  uuid = request.args.get('uuid',None,type=str)  # 查詢是否存在該 uuid  sql\_command=f"SELECT \* FROM user where uuid='{uuid}'"  data=db.get\_data(sql\_command)  number=random\_string()+str(data[0])  if len(data)==1:  # 更新註冊時間  sql\_command = f"UPDATE user SET register\_time = NOW() WHERE uuid='{uuid}'"  db.edit\_data(sql\_command)  # 使用 threading 並行執行 TIOJ 和 ZeroJudge 註冊  t1 = threading.Thread(target=registration.TIOJ\_registration, args=(number,))  t2 = threading.Thread(target=registration.ZeroJudge\_registration, args=(number,))  # 開始執行這兩個線程  t1.start()  t2.start()  # 等待這兩個線程執行完畢  t1.join()  t2.join()  # 執行新增帳戶操作  registration.add\_account(number, number, './crawler/account.json')  # 將 uuid 設置為 Null  sql\_command = f"UPDATE user SET uuid = Null WHERE uuid='{uuid}'"  db.edit\_data(sql\_command)  result='驗證成功'  db.edit\_data(sql\_command)  return render\_template('verify\_register.html',result=result)  else:  result='驗證失敗'  return render\_template('verify\_register.html',result=result)  # 忘記密碼頁面路由  @auth\_bp.route('/forget\_password' ,methods=['GET','POST'])  def forget\_password():  if request.method == "POST":  Email=request.form['Email']  # 獲取使用者名稱  user\_name=db.get\_data(f"SELECT \* FROM user where email='{Email}'")[0][1]  # 生成驗證用的 UUID  token=str(uuid.uuid4())  # 將 UUID 更新到資料庫中  db.edit\_data(f"UPDATE user SET uuid = '{token}' WHERE email='{Email}'")  html=f'http://140.131.114.141/auth/verify\_forget\_password?uuid={token}'  msg\_title = 'Forget CodeAlchemy Password'  msg\_recipients=[Email]  msg\_html =f'<p>親愛的{user\_name},</p><p>我們注意到您最近嘗試登入您的帳號時遇到了一些問題。如果您忘記了您的密碼，請不要擔心，我們很樂意協助您重設密碼。</p><p>請點擊以下連結以重設您的密碼：</p><a href="{html}">重設密碼</a><p>如果點擊上述連結無法正常工作，請複製並粘貼以下網址至您的瀏覽器中：</p><p>{html}</p><p>請注意，此連結將在收到此郵件後的24小時內有效。請盡快完成密碼重設流程。</p><p>如果您沒有請求重設密碼，請忽略此郵件。您的帳號安全是我們的首要關注。</p><p>如果您有任何疑問或需要進一步協助，請隨時回覆此郵件與我們聯繫。</p>'.encode('utf-8')  msg = Message(  subject=msg\_title,  sender = 'codealchemyproject@gmail.com',  recipients=msg\_recipients,  html=msg\_html  )  mail.send(msg)  return redirect('/auth/login')  else:  return render\_template('forget\_password.html')  # 忘記密碼驗證頁面路由  @auth\_bp.route('/verify\_forget\_password',methods=['GET','POST'])  def verify\_forget\_password():  # 獲得 UUID  uuid = request.args.get('uuid',None,type=str)  if request.method == "POST":  uuid=request.form['uuid']  # 檢查 UUID 是否存在  sql\_command=f"SELECT \* FROM user where uuid='{uuid}'"  data=db.get\_data(sql\_command)  if len(data)==1:  if request.method == "POST":  uuid=request.form['uuid']  Password=request.form['Password']  # 新密碼加密  Password=generate\_password\_hash(Password)  # 更新密碼並將 UUID 設置為 Null  sql\_command = f"UPDATE user SET password = '{Password}' WHERE uuid='{uuid}'"  db.edit\_data(sql\_command)  sql\_command = f"UPDATE user SET uuid = Null WHERE uuid='{uuid}'"  db.edit\_data(sql\_command)  result='變更成功'  return render\_template('verify\_forget\_password\_result.html',result=result)  else:  return render\_template('verify\_forget\_password.html',uuid=uuid)  else:  result='錯誤'  return render\_template('verify\_forget\_password\_result.html',result=result)  # 更改密碼頁面路由  @auth\_bp.route('/change\_password',methods=['GET','POST'])  def change\_password():  if request.method == "POST":  Email=session.get('Email')  sql\_common=f"SELECT \* FROM user where email='{Email}'"  oldPassword=request.form['oldPassword']  newPassword=request.form['newPassword']  renewPassword=request.form['renewPassword']  # 檢查新密碼是否一致  if newPassword==renewPassword:  # 新密碼加密  newPassword=generate\_password\_hash(newPassword)  # 檢查舊密碼是否正確  if check\_password\_hash(db.get\_data(sql\_common)[0][2],oldPassword):  # 更新密碼  sql\_common= f"UPDATE user SET password = '{newPassword}' WHERE email='{Email}'"  db.edit\_data(sql\_common)  return render\_template('change\_password\_success.html')  else:  result='密碼錯誤'  return render\_template('change\_password.html',result=result)  else:  result='密碼不一致'  return render\_template('change\_password.html',result=result)  else:  return render\_template('change\_password.html')  # 登出頁面路由  @auth\_bp.route('/logout')  def logout():  session.clear()  resp = make\_response(redirect('/'))  resp.set\_cookie('logged\_in','',expires=0)  resp.set\_cookie('user\_name','',expires=0)  resp.set\_cookie('user\_id','',expires=0)  return resp | |

|  |  |
| --- | --- |
| 路徑 | templates\login.html |
| 用途 | 查詢電子郵件是否已註冊 |
| 部分程式碼 | |
| <div class="container">          <div class="row">              <div class="col-sm-9 col-md-7 col-lg-5 mx-auto">                  <div class="card border-0 shadow rounded-3 my-5">                      <div class="card-body p-4 p-sm-5">                          <h2 class="card-title text-center mb-5">登入或註冊</h2>                          <form action="/auth/login" method="POST">                              <div class="form-floating mb-3">                                  <input type="email" class="form-control" id="floatingInput"                                      placeholder="name@example.com" name="Email" required>                                  <label for="floatingInput">Email address</label>                              </div>                              <div class="d-grid">                                  <button class="btn btn-primary btn-login text-uppercase fw-bold"                                      type="submit">送出</button>                              </div>                          </form>                          <hr class="my-4">                          <div class="d-grid mb-2">                              <a href="/auth/google" class="btn btn-light">                                  <!-- 嵌入 Google 圖示 SVG -->                                  <img src="../static/imgs/google-icon-logo.svg" alt="Google 圖示" style="width: 5%;">                                  使用Google登入                                </a>                          </div>                      </div>                  </div>              </div>          </div>      </div> | |

|  |  |
| --- | --- |
| 路徑 | templates\login\_password.html |
| 用途 | 登入密碼頁面 |
| 部分程式碼 | |
| <div class="container">          <div class="row">              <div class="col-sm-9 col-md-7 col-lg-5 mx-auto">                  <div class="card border-0 shadow rounded-3 my-5">                      <div class="card-body p-4 p-sm-5">                          <h2 class="card-title text-center mb-5">輸入密碼</h2>                          <form action="/auth/login\_password" method="POST">                              <div class="form-floating mb-3">                                  <input type="hidden" value="{{Email}}">                                  <input type="password" class="form-control" id="floatingPassword" placeholder="Password"                                      name="Password" required>                                  <label for="floatingPassword">Password</label>                                  <div class="invalid-feedback">                                      密碼錯誤                                  </div>                                  {% if result=='密碼錯誤' %}                                  <a href="./forget\_password" class="link-primary">忘記密碼？</a>                                  {% endif %}                              </div>                              <div class="form-check mb-3">                                  <label class="form-check-label" for="rememberPasswordCheck">                                      記住我                                  </label>                                  <input class="form-check-input" type="checkbox" value="1" id="rememberPasswordCheck"                                      name="Rememberme" checked>                              </div>                              <div class="d-grid">                                  <button class="btn btn-primary btn-login text-uppercase fw-bold"                                      type="submit">登入</button>                              </div>                          </form>                          <script>                              function handleBackendResult(result) {                                  const passwordInput = document.getElementById("floatingPassword");                                  if (result === "密碼錯誤") {                                      passwordInput.classList.add("is-invalid");                                  } else {                                      passwordInput.classList.remove("is-invalid");                                  }                              }                              const result = "{{ result }}"; // Get the result from Jinja2                              handleBackendResult(result);                          </script>                      </div>                  </div>              </div>          </div>      </div> | |

|  |  |
| --- | --- |
| 路徑 | templates\forget\_password.html |
| 用途 | 忘記密碼頁面前端 |
| 部分程式碼 | |
| <div class="container">          <div class="row">              <div class="col-sm-9 col-md-7 col-lg-5 mx-auto">                  <div class="card border-0 shadow rounded-3 my-5">                      <div class="card-body p-4 p-sm-5">                          <h2 class="card-title text-center mb-5">忘記密碼</h2>                          <form action="/auth/forget\_password" method="POST">                              <div class="form-floating mb-3">                                  <input type="email" class="form-control" id="floatingInput"                                      placeholder="name@example.com" name="Email" required>                                  <label for="floatingInput">Email address</label>                              </div>                              <div class="d-grid">                                  <button class="btn btn-primary btn-login text-uppercase fw-bold"                                      type="submit">送出</button>                              </div>                          </form>                      </div>                  </div>              </div>          </div>      </div> | |

|  |  |
| --- | --- |
| 路徑 | templates\verify\_forget\_password.html |
| 用途 | 驗證忘記密碼郵件頁面 |
| 部分程式碼 | |
| <div class="container">          <div class="row">              <div class="col-sm-9 col-md-7 col-lg-5 mx-auto">                  <div class="card border-0 shadow rounded-3 my-5">                      <div class="card-body p-4 p-sm-5">                          <h2 class="card-title text-center mb-5">重設密碼</h2>                          <form action="/auth/verify\_forget\_password" method="POST">                              <input type="hidden" value="{{uuid}}" name="uuid">                              <div class="form-floating mb-3">                                  <input type="password" class="form-control" id="floatingPassword" placeholder="Password"                                      name="Password" required>                                  <label for="floatingPassword">Password</label>                              </div>                              <div class="d-grid">                                  <button class="btn btn-primary btn-login text-uppercase fw-bold"                                      type="submit">送出</button>                              </div>                          </form>                      </div>                  </div>              </div>          </div>      </div> | |

|  |  |
| --- | --- |
| 路徑 | templates\change\_password.html |
| 用途 |  |
| 部分程式碼 | |
| <div class="container">          <div class="row">              <div class="col-sm-9 col-md-7 col-lg-5 mx-auto">                  <div class="card border-0 shadow rounded-3 my-5">                      <div class="card-body p-4 p-sm-5">                          <h2 class="card-title text-center mb-5">修改密碼</h2>                          <form action="/auth/change\_password" method="POST">                              <input type="hidden" value="{{Email}}">                              <div class="form-floating mb-3">                                  <input type="password" class="form-control" id="oldPassword" placeholder="Password"                                      name="oldPassword" required>                                  <label for="floatingPassword">Password</label>                                  <div class="invalid-feedback">                                      密碼錯誤                                  </div>                              </div>                              <div class="form-floating mb-3">                                  <input type="password" class="form-control" id="newPassword" placeholder="Password"                                      name="newPassword" required>                                  <label for="floatingPassword">New Password</label>                              </div>                              <div class="form-floating mb-3">                                  <input type="password" class="form-control" id="renewPassword" placeholder="Password"                                      name="renewPassword" required>                                  <label for="floatingPassword">Repeat New Password</label>                                  <div class="invalid-feedback">                                      密碼不一致                                  </div>                              </div>                              {{result}}                              <div class="d-grid">                                  <button class="btn btn-primary btn-login text-uppercase fw-bold"                                      type="submit">修改</button>                              </div>                      </div>                      </form>                      <script>                          function handleBackendResult(result) {                              const passwordInput = document.getElementById("oldPassword");                              const newpasswordInput = document.getElementById("newpassword");                              const renewpasswordInput = document.getElementById("renewPassword");                              if (result === "密碼錯誤") {                                  passwordInput.classList.add("is-invalid");                              }                              else if (result === '密碼不一致') {                                  renewpasswordInput.classList.add("is-invalid");                              }                              else {                                  passwordInput.classList.remove("is-invalid");                              }                          }                          const result = "{{ result }}"; // Get the result from Jinja2                          handleBackendResult(result);                      </script>                  </div>              </div>          </div>      </div> | |

|  |  |
| --- | --- |
| 路徑 | services\feedback\app.py |
| 用途 | 回饋相關功能 |
| 部分程式碼 | |
| #新增反饋表單  @feedback\_bp.route('/create/form')  def feedback\_create\_form():  return render\_template('feedback\_create\_form.html')  #新增反饋  @feedback\_bp.route('/create', methods=['POST'])  def submit\_feedback():  feedback\_content = request.form['feedback\_content']  user\_id = session['User\_id']  if feedback\_content:  #取得資料庫連線  connection = db.connection()    # 取得目前時間  created\_at = datetime.now().strftime("%Y-%m-%d %H:%M:%S")  # 將反饋資料存入資料庫  cursor = connection.cursor()  cursor.execute("INSERT INTO feedback (user\_id, content, created\_at) VALUES (%s, %s, %s)",  (user\_id, feedback\_content, created\_at))    #關閉資料庫連線  connection.commit()  connection.close()    # 使用 JavaScript 彈跳視窗顯示成功消息  success\_message = "送出成功!"  return f'''  <script>  alert("{success\_message}");  window.location.replace("/feedback/create/form");  </script>  '''  else:  # 使用 JavaScript 彈跳視窗顯示失敗消息  error\_message = "送出失敗!"  return f'''  <script>  alert("{error\_message}");  window.location.replace("/feedback/create/form");  </script>  '''    #反饋紀錄  @feedback\_bp.route('/feedback\_history')  def feedback\_history():  #取得資料庫連線  connection = db.connection()    #產生執行sql命令的物件, 再執行sql  cursor = connection.cursor()  #取得傳入參數, 執行sql命令並取回資料  user\_id = session.get('User\_id')  cursor.execute('SELECT \* FROM feedback WHERE user\_id=%s ORDER BY created\_at DESC', (user\_id,))  feedback\_history = cursor.fetchall()  #關閉連線  connection.close()    #渲染網頁  return render\_template('feedback\_history.html', feedback\_history=feedback\_history)  #管理者反饋紀錄  @feedback\_bp.route('/admin\_dashboard')  def admin\_dashboard():  # 取得當前頁碼，默認為 1  page = request.args.get('page', 1, type=int)  per\_page = 5  #取得資料庫連線  connection = db.connection()    #產生執行sql命令的物件, 再執行sql  cursor = connection.cursor()  # 查詢所有反饋的總數量  cursor.execute('SELECT COUNT(\*) FROM feedback')  total\_feedback = cursor.fetchone()[0]  # 計算總頁數  total\_pages = (total\_feedback + per\_page - 1) // per\_page  # 計算分頁所需的偏移量  offset = (page - 1) \* per\_page  # 查詢特定頁面的反饋資料，按提交時間降序排列，並進行表連接  query = '''  SELECT feedback.feedback\_id, user.user\_name, feedback.content, feedback.created\_at, feedback.reply  FROM feedback  JOIN user ON feedback.user\_id = user.user\_id  ORDER BY feedback.created\_at DESC  LIMIT %s OFFSET %s  '''  cursor.execute(query, (per\_page, offset))  feedback = cursor.fetchall()    #關閉連線  connection.close()  print(f'Page: {page}, Total Pages: {total\_pages},{total\_feedback}')  return render\_template('admin\_dashboard.html', feedback=feedback, page=page, total\_pages=total\_pages)  #回覆反饋表單  @feedback\_bp.route('/reply\_feedback/form', methods=['GET'])  def reply\_feedback\_form():  #取得資料庫連線  connection = db.connection()    #產生執行sql命令的物件, 再執行sql  cursor = connection.cursor()    feedback\_id = request.args.get('feedback\_id')  query = '''  SELECT feedback.feedback\_id, user.user\_name, feedback.content  FROM feedback  JOIN user ON feedback.user\_id = user.user\_id  WHERE feedback.feedback\_id=%s  '''  cursor.execute(query, (feedback\_id,))  feedback = cursor.fetchone()    #渲染網頁  return render\_template('reply\_feedback\_form.html', feedback=feedback)  #回覆反饋  @feedback\_bp.route('/reply\_feedback', methods=['POST'])  def reply\_feedback():  #取得資料庫連線  connection = db.connection()    #產生執行sql命令的物件, 再執行sql  cursor = connection.cursor()  feedback\_id = request.form['feedback\_id']  reply\_content = request.form['reply\_content']  if feedback\_id and reply\_content:  # 更新資料庫中的回覆欄位  cursor.execute('UPDATE feedback SET reply=%s WHERE feedback\_id=%s', (reply\_content, feedback\_id))  connection.commit()  #關閉連線  connection.close()  # 使用 JavaScript 彈跳視窗顯示成功消息  success\_message = "送出成功!"  return f'''  <script>  alert("{success\_message}");  window.location.replace("/feedback/admin\_dashboard");  </script>  '''  else:  # 使用 JavaScript 彈跳視窗顯示失敗消息  error\_message = "送出失敗!"  return f'''  <script>  alert("{error\_message}");  window.location.replace("/feedback/admin\_dashboard");  </script>  ''' | |

|  |  |
| --- | --- |
| 路徑 | templates\feedback\_create\_form.html |
| 用途 | 用戶回饋前端 |
| 部分程式碼 | |
| <div class="container">          <div class="card border-0 shadow rounded-3 my-5">              <div class="card-body p-4 p-sm-5">                  <h2>請提供您的反饋:</h2>                  <form action="/feedback/create" method="post">                      <textarea class="form-control" name="feedback\_content" rows="5" placeholder="在此輸入您的反饋"></textarea>                      <div class="btn-row">                          <button type="submit" class="btn btn-primary">提交反饋</button>                          <a href="/feedback/feedback\_history" class="btn btn-primary">反饋紀錄</a>                      </div>                  </form>              </div>          </div>      </div> | |

|  |  |
| --- | --- |
| 路徑 | templates\feedback\_history.html |
| 用途 | 回饋紀錄 |
| 部分程式碼 | |
| <div class="container">          {% if feedback\_history %}              <table>                  <tr>                      <th>反饋內容</th>                      <th>提交時間</th>                      <th>回覆</th>                  </tr>                  {% for feedback in feedback\_history %}                      <tr>                          <td>{{ feedback[2] }}</td>                          <td>{{ feedback[3] }}</td>                          <td>{{ feedback[4] }}</td>                      </tr>                  {% endfor %}              </table>          {% else %}              <p>沒有反饋紀錄</p>          {% endif %}      </div> | |

|  |  |
| --- | --- |
| 路徑 | templates\admin\_dashboard.html |
| 用途 | 管理回饋表單 |
| 部分程式碼 | |
| <div class="feedback-container">          <div class="row">              <div class="col-md-7 mx-auto">                  {% for feedback in feedback %}                  <div class="feedback-item card border-0 shadow">                      <p><strong>使用者名稱:</strong> {{ feedback[1] }}</p>                      <p><strong>反饋內容:</strong> {{ feedback[2] }}</p>                      <p><strong>提交時間:</strong> {{ feedback[3] }}</p>                      <p><strong>回覆內容:</strong> {{ feedback[4] }}</p>                      <button class="reply-button"                          onclick="location.href='/feedback/reply\_feedback/form?feedback\_id={{ feedback[0] }}'">回覆</button>                  </div>                  {% endfor %}              </div>          </div>      </div>      <div class="pagination">          {% if page > 1 %}          <a href="{{ url\_for('feedback\_bp.admin\_dashboard', page=page-1) }}">上一頁</a>          {% endif %}          {% for p in range(1, total\_pages + 1) %}          {% if p == page %}          <p>{{ p }}</p>          {% else %}          <a href="{{ url\_for('feedback\_bp.admin\_dashboard', page=p) }}">{{ p }}</a>          {% endif %}          {% endfor %}          {% if page < total\_pages %} <a href="{{ url\_for('feedback\_bp.admin\_dashboard', page=page+1) }}">下一頁</a>              {% endif %}      </div> | |

|  |  |
| --- | --- |
| 路徑 | templates\reply\_feedback\_form.html |
| 用途 | 回復回饋表單 |
| 部分程式碼 | |
| <div class="reply-form">          <form action="/feedback/reply\_feedback" method="post">              <p><strong>使用者名稱:</strong> {{ feedback[1] }}</p>              <p><strong>反饋內容:</strong> {{ feedback[2] }}</p>              <input type="hidden" name="feedback\_id" value="{{ feedback[0] }}">              <label for="reply\_content">回覆:</label><br>              <textarea name="reply\_content" rows="5" cols="50"></textarea><br>              <input type="submit" value="回覆">          </form>      </div> | |

|  |  |
| --- | --- |
| 路徑 | services\user\app.py |
| 用途 | 用戶個人資料相關功能 |
| 部分程式碼 | |
| # 獲取當前的目錄路徑  current\_directory = os.path.dirname(os.path.realpath(\_\_file\_\_))  # 獲取項目的跟目錄路徑  root\_directory = os.path.abspath(os.path.join(current\_directory, '..', '..'))  # 建立相對路徑  UPLOAD\_FOLDER = os.path.join(root\_directory, 'static', 'user\_icon')  ALLOWED\_EXTENSIONS = set(['pdf', 'png', 'jpg', 'jpeg', 'gif'])  # 定義函式來檢查文件擴展名  def allowed\_file(filename):      return '.' in filename and filename.rsplit('.', 1)[1].lower() in ALLOWED\_EXTENSIONS  @user\_bp.route('/user\_data', methods=['GET', 'POST'])  def user\_data():      # 從 session 中獲取用戶的 Email      User\_id = session.get('User\_id') or request.cookies.get('user\_id')      # 根據 Email 查詢用戶數據      sql\_command = f"SELECT \* FROM user where user\_id='{User\_id}'"      data = db.get\_data(sql\_command)      # 提取用戶數據      User\_id = data[0][0]      User\_name = data[0][1]      Google\_id = data[0][3]      Email = data[0][4]      img = data[0][5]      register\_time = data[0][8]      # 取得個人解題紀錄      sql\_command = f"SELECT \* FROM `answer record` where user\_id='{User\_id}';"      data = db.get\_data(sql\_command)      answer\_record=data      # 取得個人收藏紀錄      sql\_command = f"SELECT collection.user\_id,collection.problem\_id,problem.title,problem.difficulty FROM `113-CodeAlchemy`.collection left join `113-CodeAlchemy`.problem on collection.problem\_id=problem.problem\_id where collection.user\_id={User\_id};"      data = db.get\_data(sql\_command)      collection=data      print(collection)      # 如果是 POST 請求，處理用戶上傳的文件      if request.method == "POST":          User\_name=request.form['User\_name']          Email=request.form['Email']          filename=img          if 'file' not in request.files:              # 如果沒有選擇文件，則不執行文件上傳操作              return redirect(request.url)          file = request.files['file']          if file and allowed\_file(file.filename):              # 文件名安全化處理並加上唯一值              filename = str(uuid.uuid4()) + '\_'+secure\_filename(file.filename)              # 將文件保存到指定目錄              file.save(os.path.join(UPLOAD\_FOLDER, filename))              # 更新用戶資料              sql\_command = f"UPDATE user SET user\_name='{User\_name}',email='{Email}',image='../static/user\_icon/{filename}' WHERE user\_id='{User\_id}'"          else:              sql\_command = f"UPDATE user SET user\_name='{User\_name}',email='{Email}' WHERE user\_id='{User\_id}'"          db.edit\_data(sql\_command)          # 重定向到用戶資料頁面          return redirect(url\_for('user.user\_data'))      else:          # 如果是 GET 請求，將用戶數據傳遞給模板並返回相應的頁面          return render\_template('./user\_data.html', User\_id=User\_id, User\_name=User\_name, Google\_id=Google\_id, Email=Email, img=img, register\_time=register\_time,answer\_record=answer\_record,collection=collection) | |

|  |  |
| --- | --- |
| 路徑 | templates\user\_data.html |
| 用途 | 用戶個人資料頁面 |
| 部分程式碼 | |
| <div class="container-md">          <ul class="nav nav-tabs nav-justified" id="myTab" role="tablist">              <li class="nav-item" role="presentation">                  <button class="nav-link active" data-bs-toggle="tab" data-bs-target="#menu1"                      aria-selected="true">個人資料</button>              </li>              <li class="nav-item" role="presentation">                  <button class="nav-link" data-bs-toggle="tab" data-bs-target="#menu2"                      aria-selected="false">個人解題紀錄</button>              </li>              <li class="nav-item" role="presentation">                  <button class="nav-link" data-bs-toggle="tab" data-bs-target="#menu3" type="button" role="tab"                      aria-controls="contact" aria-selected="false">收藏紀錄</button>              </li>          </ul>          <div class="tab-content" id="myTabContent">              <!-- 顯示數據 -->              <div class="tab-pane fade show active" id="menu1" role="tabpanel">                  <div class="col-sm-9 col-md-7 col-lg-9 mx-auto">                      <form action="/user/user\_data" method="post" enctype="multipart/form-data">                          <div class="row mb-3">                              <label for="inputEmail3" class="col-sm-2 col-form-label">個人檔案圖片</label>                              <!-- 個人檔案圖片 -->                              {% if img %}                              <!-- 如果有圖片 -->                              <div class="card border-0 w-25">                                  <img src="{{ img }}" style="width: 96px;height: 96px;" class="rounded-circle">                              </div>                              <!-- 顯示圖片 -->                              {% else %}                              <!-- 如果沒有圖片 -->                              {% endif %}                          </div>                          <div class="input-group mb-3">                              <label for="inputEmail3" class="col-sm-2 col-form-label">上傳個人檔案圖片</label>                              <!-- 上傳個人檔案圖片標籤 -->                              <input type="file" class="form-control" id="inputGroupFile01" name="file" accept="image/\*">                              <!-- 上傳圖片輸入框 -->                          </div>                          <div class="row mb-3">                              <label for="inputEmail3" class="col-sm-2 col-form-label">使用者ID</label>                              <!-- 使用者ID標籤 -->                              <div class="col-sm-10">                                  <input type="text" readonly class="form-control" id="User\_id" value="{{ User\_id }}">                                  <!-- 顯示使用者ID -->                              </div>                          </div>                          <div class="row mb-3">                              <label class="col-sm-2 col-form-label">使用者名稱</label>                              <!-- 使用者名稱標籤 -->                              <div class="col-sm-10">                                  <input type="text" class="form-control" name="User\_name" value="{{User\_name}}">                                  <!-- 顯示使用者名稱 -->                              </div>                          </div>                          <div class="row mb-3">                              <label for="inputEmail3" class="col-sm-2 col-form-label">Email</label>                              <!-- Email標籤 -->                              <div class="col-sm-10">                                  <input type="email" class="form-control" name="Email" value="{{Email}}">                                  <!-- 顯示Email -->                              </div>                          </div>                          <div class="row mb-3">                              <label class="col-sm-2 col-form-label">連結到Google</label>                              <!-- 連結到Google標籤 -->                              <div class="col-sm-10">                                  {% if Google\_id %}                                  <!-- 如果已連結Google -->                                  <label for="inputEmail3" class="col-sm-2 col-form-label">已連結到Google</label>                                  <!-- 顯示已連結到Google -->                                  {% else %}                                  <!-- 如果尚未連結Google -->                                  <input type="radio" class="btn-check" name="options-outlined" id="success-outlined"                                      autocomplete="off">                                  <a href="/auth/google" class="btn btn-light">                                      <!-- 嵌入 Google 圖示 SVG -->                                      <img src="../static/imgs/google-icon-logo.svg" alt="Google 圖示" style="width: 3%;">                                      使用Google登入                                      <!-- 使用Google登入按鈕 -->                                  </a>                                  {% endif %}                              </div>                          </div>                          <div class="row mb-3">                              <label for="inputEmail3" class="col-sm-2 col-form-label">密碼</label>                              <!-- 密碼標籤 -->                              <div class="col-sm-10">                                  <input type="radio" class="btn-check" name="options-outlined" id="success-outlined"                                      autocomplete="off">                                  <!-- 變更密碼按鈕 -->                                  <a href="/auth/change\_password" class="btn btn-primary">                                      變更密碼                                  </a>                              </div>                          </div>                          <button type="submit" class="btn btn-primary">儲存</button>                          <!-- 儲存按鈕 -->                      </form>                  </div>              </div>              <div class="tab-pane fade" id="menu2">                  <div class="table-responsive">                      <table class="table">                          <thead>                              <tr>                                  <th scope="col">題號</th>                                  <th scope="col">語言</th>                                  <th scope="col">執行時間</th>                                  <th scope="col">花費記憶體</th>                                  <th scope="col">上傳時間</th>                                  <th scope="col">結果</th>                              </tr>                          </thead>                          <tbody>                              {% for each\_answer\_record in answer\_record %}                              <tr class="">                                  <td>{{each\_answer\_record[2]}}</td>                                  <td>{{each\_answer\_record[4]}}</td>                                  <td>{{each\_answer\_record[5]}}</td>                                  <td>{{each\_answer\_record[6]}}</td>                                  <td>{{each\_answer\_record[7]}}</td>                                  {% if each\_answer\_record[3] == 'Accepted'%}                                  <td><p class="text-success">通過</p></td>                                  {% else %}                                  <td><p class="text-danger">未通過</p></td>                                  {% endif %}                              </tr>                              {% endfor %}                          </tbody>                      </table>                  </div>              </div>              <div class="tab-pane fade" id="menu3">                  <div class="table">                      <table class="table table-hover">                          <thead>                              <tr>                                  <th scope="col">編號</th>                                  <th scope="col">名稱</th>                                  <th scope="col">難度</th>                              </tr>                          </thead>                          <tbody>                              {% for data in collection %}                              <tr onclick="window.location='../problem?problem\_id={{data[1]}}'">                                  <td>{{data[1]}}</td>                                  <td>{{data[2]}}</td>                                  <td>{{data[3]}}</td>                              </tr>                              {% endfor %}                          </tbody>                      </table>                  </div>              </div>          </div>      </div> | |

|  |  |
| --- | --- |
| 路徑 | services\manager\app.py |
| 用途 | 管理者相關功能 |
| 部分程式碼 | |
| # 新增題目  @manager\_bp.route('add\_problem',methods=["GET","POST"])  def add\_problem():      try:          if session['Permission']=='Manager':              if request.method=="POST":                  #給新題號                  sql\_command="SELECT \* FROM `113-CodeAlchemy`.problem where problem\_id like 'CAOJ%' order by problem\_id desc;"                  old\_no=db.get\_data(sql\_command)[0][0].split('-')[1]                  new\_no=int(old\_no)+1                  #處理新增題目內容                  data=request.form                  title=data.get('title')                  content=data.get('content')                  enter\_description=data.get('enter\_description')                  output\_description=data.get('output\_description')                  example\_input=data.get('example\_input')                  example\_output=data.get('example\_output')                  difficulty=data.get('difficulty')                  tag=data.get('tag')                  video\_id=data.get('video\_id')                  sql\_command=f"INSERT INTO `113-CodeAlchemy`.`problem` (`problem\_id`, `title`, `content`, `enter\_description`, `output\_description`, `example\_input`, `example\_output`, `difficulty`, `tag`, `video\_id` ,`solved`, `submission`, `update\_time`) VALUES ('CAOJ-{new\_no}', '{title}', '{content}', '{enter\_description}', '{output\_description}', '{example\_input}' , '{example\_output}', '{difficulty}', '{tag}', '{video\_id}', '0', '0', '{datetime.now()}');"                  db.edit\_data(sql\_command)                  return redirect('/')              else:                  return render\_template('./add\_problem.html')      except:          return redirect('/')  # 修改題目  @manager\_bp.route('/problem',methods=['GET',"POST"])  def problem():      try:          if session['Permission']=='Manager':              if request.method=="POST":                  data=request.form                  problem\_id=data.get('problem\_id')                  title=data.get('title')                  content=data.get('content')                  enter\_description=data.get('enter\_description')                  output\_description=data.get('output\_description')                  example\_input=data.get('example\_input')                  example\_output=data.get('example\_output')                  difficulty=data.get('difficulty')                  tag=data.get('tag')                  video\_id=data.get('video\_id')                  sql\_problem\_command=f"UPDATE problem SET title = '{title}',content = '{content}',enter\_description = '{enter\_description}',output\_description = '{output\_description}',example\_input = '{example\_input}',example\_output = '{example\_output}' ,difficulty='{difficulty}' ,tag='{tag}' ,video\_id='{video\_id}' WHERE problem\_id = '{problem\_id}';"                  db.edit\_data(sql\_problem\_command)                  sql\_command=f"SELECT \* FROM problem where problem\_id='{problem\_id}'"                  problem\_data=db.get\_data(sql\_command)                  example\_inputs = problem\_data[0][5].split('|||')                  example\_outputs = problem\_data[0][6].split('|||')                  tag = problem\_data[0][8]                  video\_id = problem\_data[0][9]                  return render\_template('./manager\_problem.html',data=problem\_data,example\_inputs=example\_inputs,example\_outputs=example\_outputs,difficulty=difficulty,tag=tag,video\_id=video\_id)              else:                  problem\_id = request.args.get('problem\_id',type=str)                  sql\_command=f"SELECT \* FROM problem where problem\_id='{problem\_id}'"                  problem\_data=db.get\_data(sql\_command)                  example\_inputs = problem\_data[0][5].split('|||')                  example\_outputs = problem\_data[0][6].split('|||')                  difficulty = problem\_data[0][7]                  tag= problem\_data[0][8]                  video\_id = problem\_data[0][9]                  return render\_template('./manager\_problem.html',data=problem\_data,example\_inputs=example\_inputs,example\_outputs=example\_outputs,difficulty=difficulty,tag=tag,video\_id=video\_id)      except:          return redirect('/')  # 管理使用者  @manager\_bp.route('/user', methods=['GET','POST'])  def user():      try:          if session["Permission"]=='Manager':              if request.method=='GET':                  sql\_command=f"SELECT \* FROM `113-CodeAlchemy`.user;"                  user\_data=db.get\_data(sql\_command)                  return render\_template('./manager\_user.html',data=user\_data )              else:                  new\_data=request.form                  new\_data=list(new\_data.items())                  sql\_command=f"SELECT \* FROM `113-CodeAlchemy`.user;"                  user\_data=db.get\_data(sql\_command)                  for i in range(len(new\_data)):                      if new\_data[i][1]!=user\_data[i][6]:                          sql\_command=f"UPDATE `113-CodeAlchemy`.`user` SET `permission` = '{new\_data[i][1]}' WHERE (`user\_id` = '{user\_data[i][0]}');"                          db.edit\_data(sql\_command)                  sql\_command=f"SELECT \* FROM `113-CodeAlchemy`.user;"                  user\_data=db.get\_data(sql\_command)                  return render\_template('./manager\_user.html',data=user\_data)      except:          return redirect('/')  # 刪除題目  @manager\_bp.route('/delete',methods=['GET'])  def delete():      try:          if session["Permission"]=='Manager':              problem\_id = request.args.get('problem\_id',type=str)              sql\_command=f"DELETE FROM problem where problem\_id='{problem\_id}'"              db.edit\_data(sql\_command)              return redirect('/')      except:          return redirect('/') | |

|  |  |
| --- | --- |
| 路徑 | templates\add\_problem.html |
| 用途 | 新增題目頁面 |
| 部分程式碼 | |
| <div class="container">          <form action="/manager/add\_problem" method="post">              <div class="row justify-content-center align-items-center g-2 mt-4">                  <h3 class="mt-4">題目標題:</h3>                  <div class="mb-3">                      <input type="text" class="form-control" name="title" id="" aria-describedby="basic-addon2"                          placeholder="" />                  </div>              </div>              <div class="          row justify-content-center align-items-center g-2">                  <h3 class="mt-4">題目說明:</h3>                  <div class="mb-3">                      <div class="mb-3">                          <textarea class="form-control" name="content" id="" rows="3"></textarea>                      </div>                  </div>              </div>              <div class="row justify-content-center align-items-center g-2"></div>              <h3 class="mt-4">輸入說明:</h3>              <div class="mb-3">                  <textarea class="form-control" name="enter\_description" id="" rows="3"></textarea>              </div>              <h3 class="mt-4">輸出說明:</h3>              <div class="mb-3">                  <textarea class="form-control" name="output\_description" id="" rows="3"></textarea>              </div>              <h3 class="mt-4">範例輸入:</h3>              <div class="mb-3">                  <textarea class="form-control" name="example\_input" id="" rows="3"></textarea>              </div>              <h3 class="mt-4">範例輸出:</h3>              <div class="mb-3">                  <textarea class="form-control" name="example\_output" id="" rows="3"></textarea>              </div>              <h3 class="mt-4">難度<br></h3>              <div class="mb-3">                  <select class="form-select form-select-lg" name="difficulty" id="">                      <option value="easy" {% if (difficulty=='easy' ) %}selected{% endif %}>Easy</option>                      <option value="middle" {% if (difficulty=="middle" ) %}selected{% endif %}>Middle</option>                      <option value="hard" {% if (difficulty=="hard" ) %}selected{% endif %}>Hard</option>                  </select>              </div>              <h3 class="mt-4">標籤</h3>              <div class="mb-3">                  <input type="text" class="form-control" name="tag" id="" aria-describedby="標籤" placeholder="" />              </div>              <h3 class="mt-4">影片ID</h3>              <div class="mb-3">                  <input type="text" class="form-control" name="video\_id" id="" aria-describedby="影片ID" placeholder="" />              </div>              <button type="button" class="btn btn-success" id="save\_btn" data-bs-toggle="modal"                  data-bs-target="#save\_modal"><svg xmlns="http://www.w3.org/2000/svg" width="16" height="16"                      fill="currentColor" class="bi bi-floppy2-fill" viewBox="0 0 16 16">                      <path d="M12 2h-2v3h2z" />                      <path                          d="M1.5 0A1.5 1.5 0 0 0 0 1.5v13A1.5 1.5 0 0 0 1.5 16h13a1.5 1.5 0 0 0 1.5-1.5V2.914a1.5 1.5 0 0 0-.44-1.06L14.147.439A1.5 1.5 0 0 0 13.086 0zM4 6a1 1 0 0 1-1-1V1h10v4a1 1 0 0 1-1 1zM3 9h10a1 1 0 0 1 1 1v5H2v-5a1 1 0 0 1 1-1" />                  </svg>儲存</button>          </form>          <div class="modal fade" id="save\_modal" tabindex="-1" aria-labelledby="exampleModalLabel" aria-hidden="true">              <div class="modal-dialog">                  <div class="modal-content">                      <div class="modal-header">                          <h5 class="modal-title" id="exampleModalLabel">儲存確認</h5>                          <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>                      </div>                      <div class="modal-body">                          確定要儲存變更嗎？                      </div>                      <div class="modal-footer">                          <button type="button" class="btn btn-secondary" data-bs-dismiss="modal">取消</button>                          <button type="button" class="btn btn-primary" id="confirm\_save">儲存變更</button>                      </div>                  </div>              </div>          </div>      </div>      <script>          document.getElementById('confirm\_save').addEventListener('click', function () {              document.querySelector('form').submit();          });      </script>      </div>      <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0/dist/js/bootstrap.bundle.min.js"          integrity="sha384-p34f1UUtsS3wqzfto5wAAmdvj+osOnFyQFpp4Ua3gs/ZVWx6oOypYoCJhGGScy+8"          crossorigin="anonymous"></script> | |

|  |  |
| --- | --- |
| 路徑 | templates\manager\_problem.html |
| 用途 | 管理題目頁面 |
| 部分程式碼 | |
| <div class="container-md">          <div class="row">              <form action="/manager/problem" method="POST">                  <div class="col">                      <div class="row">                          <h3 class="mt-4">題目代號:<br></h3>                          <h1 class="display-6 mt-4">                              <div class="input-group">                                  <input type="text" class="form-control" placeholder="題目代號" aria-label="題目代號"                                      aria-describedby="basic-addon2" value="{{data[0][0]}}" name="problem\_id" readonly>                              </div>                          </h1>                          <h3 class="mt-4">題目標題:<br></h3>                          <h1 class="display-6 mt-4">                              <div class="input-group">                                  <input type="text" class="form-control" placeholder="題目名稱" aria-label="題目名稱"                                      aria-describedby="basic-addon2" value="{{data[0][1]}}" name="title">                              </div>                          </h1>                          <div class="col-12">                              <h3 class="mt-4">題目說明:<br></h3>                              <div class="input-group">                                  <textarea class="form-control" placeholder="題目內容" aria-label="題目內容" id="content"                                      aria-describedby="basic-addon2" name="content">{{data.0.2}}</textarea>                              </div>                              <!-- textarea自動適應題目說明大小 -->                              <script>                                  const textarea = document.getElementById('content');                                  textarea.addEventListener('input', function () {                                      this.style.height = 'auto';                                      this.style.height = `${this.scrollHeight}px`;                                  });                                  textarea.dispatchEvent(new Event('input'));                              </script>                              <h3 class="mt-4">輸入說明:<br></h3>                              <div class="input-group">                                  <textarea class="form-control" placeholder="題目內容" aria-label="題目內容"                                      aria-describedby="basic-addon2" name="enter\_description">{{data.0.3}}</textarea>                              </div>                              <h3 class="mt-4">輸出說明:<br></h3>                              <div class="input-group">                                  <textarea class="form-control" placeholder="題目內容" aria-label="題目內容"                                      aria-describedby="basic-addon2" name="output\_description">{{data.0.4}}</textarea>                              </div>                              <h3 class="mt-4">範例輸入:<br></h3>                              <div class="input-group">                                  <textarea class="form-control" placeholder="題目內容" aria-label="題目內容"                                      aria-describedby="basic-addon2" name="example\_input">{{data.0.5}}</textarea>                              </div>                              <h3 class="mt-4">範例輸出:<br></h3>                              <div class="input-group">                                  <textarea class="form-control" placeholder="題目內容" aria-label="題目內容"                                      aria-describedby="basic-addon2" name="example\_output">{{data.0.6}}</textarea>                              </div>                              <h3 class="mt-4">難度<br></h3>                              <div class="mb-3">                                  <select class="form-select form-select-lg" name="difficulty" id="">                                      <option value="easy" {% if (difficulty=='easy' ) %}selected{% endif %}>Easy</option>                                      <option value="middle" {% if (difficulty=="middle" ) %}selected{% endif %}>Middle                                      </option>                                      <option value="hard" {% if (difficulty=="hard" ) %}selected{% endif %}>Hard</option>                                  </select>                              </div>                              <h3 class="mt-4">標籤</h3>                              <div class="mb-3">                                  <input type="text" class="form-control" name="tag" id="" aria-describedby="標籤"                                      placeholder="" value="{{data.0.8}}" />                              </div>                              <h3 class="mt-4">影片ID</h3>                              <div class="mb-3">                                  <input type="text" class="form-control" name="video\_id" id="" aria-describedby="影片ID"                                      placeholder="" value="{{data.0.9}}" />                              </div>                          </div>                      </div>                  </div>                  <div class="col mt-5">                      <div class="row d-flex justify-content-center">                          <button type="button" class="btn btn-primary btn-success" id="save\_btn" data-bs-toggle="modal"                              data-bs-target="#save\_modal">                              <svg xmlns="http://www.w3.org/2000/svg" width="16" height="16" fill="currentColor"                                  class="bi bi-floppy2-fill" viewBox="0 0 16 16">                                  <path d="M12 2h-2v3h2z" />                                  <path                                      d="M1.5 0A1.5 1.5 0 0 0 0 1.5v13A1.5 1.5 0 0 0 1.5 16h13a1.5 1.5 0 0 0 1.5-1.5V2.914a1.5 1.5 0 0 0-.44-1.06L14.147.439A1.5 1.5 0 0 0 13.086 0zM4 6a1 1 0 0 1-1-1V1h10v4a1 1 0 0 1-1 1zM3 9h10a1 1 0 0 1 1 1v5H2v-5a1 1 0 0 1 1-1" />                              </svg>儲存</button>                      </div>                  </div>              </form>              <div class="modal fade" id="save\_modal" tabindex="-1" aria-labelledby="exampleModalLabel"                  aria-hidden="true">                  <div class="modal-dialog">                      <div class="modal-content">                          <div class="modal-header">                              <h5 class="modal-title" id="exampleModalLabel">儲存確認</h5>                              <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>                          </div>                          <div class="modal-body">                              確定要儲存變更嗎？                          </div>                          <div class="modal-footer">                              <button type="button" class="btn btn-secondary" data-bs-dismiss="modal">取消</button>                              <button type="button" class="btn btn-primary" id="confirm\_save">儲存變更</button>                          </div>                      </div>                  </div>              </div>          </div>      </div>      <script>          document.getElementById('confirm\_save').addEventListener('click', function () {              document.querySelector('form').submit();          });      </script>      <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0/dist/js/bootstrap.bundle.min.js"          integrity="sha384-p34f1UUtsS3wqzfto5wAAmdvj+osOnFyQFpp4Ua3gs/ZVWx6oOypYoCJhGGScy+8"          crossorigin="anonymous"></script> | |

|  |  |
| --- | --- |
| 路徑 | templates\manager\_user.html |
| 用途 | 管理使用者頁面 |
| 部分程式碼 | |
| <div class="container">          <form action="/manager/user" method="POST">              <div class="table-responsive">                  <table class="table">                      <thead>                          <tr>                              <th scope="col">使用者ID</th>                              <th scope="col">使用者名稱</th>                              <th scope="col">權限</th>                          </tr>                      </thead>                      <tbody>                          {% for item in data%}                          <tr class="">                              <td scope="row">{{item[0]}}</td>                              <td>{{item[1]}}</td>                              <td>                                  <div class="">                                      <select class="form-select" name="{{ item[0] }}" id="">                                          <option value="Default user" {% if item[6]=='Default user' %} selected {% endif                                              %}>                                              使用者                                          </option>                                          <option value="Manager" {% if item[6]=='Manager' %} selected {% endif %}>                                              管理者                                          </option>                                          <option value="Banner" {% if item[6]=='Banner' %} selected {% endif %}>                                              封禁                                          </option>                                      </select>                                  </div>                              </td>                          </tr>                          {% endfor %}                      </tbody>                  </table>                  <button type="button" class="btn btn-primary btn-success" id="save\_btn" data-bs-toggle="modal"                      data-bs-target="#save\_modal">                      <svg xmlns="http://www.w3.org/2000/svg" width="16" height="16" fill="currentColor"                          class="bi bi-floppy2-fill" viewBox="0 0 16 16">                          <path d="M12 2h-2v3h2z" />                          <path                              d="M1.5 0A1.5 1.5 0 0 0 0 1.5v13A1.5 1.5 0 0 0 1.5 16h13a1.5 1.5 0 0 0 1.5-1.5V2.914a1.5 1.5 0 0 0-.44-1.06L14.147.439A1.5 1.5 0 0 0 13.086 0zM4 6a1 1 0 0 1-1-1V1h10v4a1 1 0 0 1-1 1zM3 9h10a1 1 0 0 1 1 1v5H2v-5a1 1 0 0 1 1-1" />                      </svg>儲存</button>          </form>          <div class="modal fade" id="save\_modal" tabindex="-1" aria-labelledby="exampleModalLabel" aria-hidden="true">              <div class="modal-dialog">                  <div class="modal-content">                      <div class="modal-header">                          <h5 class="modal-title" id="exampleModalLabel">儲存確認</h5>                          <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>                      </div>                      <div class="modal-body">                          確定要儲存變更嗎？                      </div>                      <div class="modal-footer">                          <button type="button" class="btn btn-secondary" data-bs-dismiss="modal">取消</button>                          <button type="button" class="btn btn-primary" id="confirm\_save">儲存變更</button>                      </div>                  </div>              </div>          </div>      </div>      <script>          document.getElementById('confirm\_save').addEventListener('click', function () {              document.querySelector('form').submit();          });      </script>      </div>      <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0/dist/js/bootstrap.bundle.min.js"          integrity="sha384-p34f1UUtsS3wqzfto5wAAmdvj+osOnFyQFpp4Ua3gs/ZVWx6oOypYoCJhGGScy+8"          crossorigin="anonymous"></script> | |

|  |  |
| --- | --- |
| 路徑 | crawler\ add\_problem.py |
| 用途 | 管理員上傳新的題目 |
| 部分程式碼 | |
| # 引入所需的模組和套件  from datetime import datetime  import re  from selenium import webdriver  from selenium.webdriver.chrome.service import Service  from webdriver\_manager.chrome import ChromeDriverManager  from selenium.webdriver.common.by import By  from selenium.webdriver.support.ui import WebDriverWait  from selenium.webdriver.support import expected\_conditions as EC  from time import sleep  import os  import traceback  def CodeAlchemy\_add\_problem(problem\_id, title, content, enter\_description, output\_description, example\_input, example\_output):      main\_url = 'http://123.192.165.145:8081/Login'      prefs = {'profile.default\_content\_setting\_values': {'notifications': 2}}      driver = None  # 初始化 driver 變數      try:          s = Service(ChromeDriverManager().install())          chrome\_options = webdriver.ChromeOptions()          chrome\_options.add\_experimental\_option('prefs', prefs)          chrome\_options.add\_experimental\_option('excludeSwitches', ['enable-logging'])          chrome\_options.add\_argument("disable-infobars")          os.environ["WDM\_ARCH"] = "64"          driver = webdriver.Chrome(service=s, options=chrome\_options)          driver.maximize\_window()          wait\_max = 10          username = 'zero'          password = '!@#$zerojudge'          driver.get(main\_url)          try:              login\_area = WebDriverWait(driver, wait\_max).until(EC.presence\_of\_element\_located(                  (By.CLASS\_NAME, 'col-md-4.text-center')))              login\_area = WebDriverWait(login\_area, wait\_max).until(                  EC.presence\_of\_element\_located((By.CLASS\_NAME, 'form-horizontal')))              sleep(2)              input\_username = WebDriverWait(login\_area, wait\_max).until(                  EC.presence\_of\_element\_located((By.ID, 'account')))              input\_username.send\_keys(username)              sleep(2)              input\_password = WebDriverWait(login\_area, wait\_max).until(                  EC.presence\_of\_element\_located((By.ID, 'passwd')))              input\_password.send\_keys(password)              sleep(3)              for i in range(300):                  sleep(1)                  try:                      btn\_login = WebDriverWait(login\_area, wait\_max).until(                          EC.presence\_of\_element\_located((By.CSS\_SELECTOR, 'button.btn.btn-primary')))                      btn\_login.click()                      break                  except:                      pass              else:                  raise BaseException              sleep(5)              if driver.current\_url != 'http://123.192.165.145:8081/':                  raise BaseException              else:                  print("登入成功")              driver.get('http://123.192.165.145:8081/InsertProblem')              # 等待標題輸入框出現              title\_input = WebDriverWait(driver, wait\_max).until(                  EC.presence\_of\_element\_located((By.NAME, 'title')))              title\_input.clear()              title\_input.send\_keys(title)              # 切換到 iframe 並輸入 content              WebDriverWait(driver, wait\_max).until(                  EC.presence\_of\_element\_located((By.ID, 'mce\_0\_ifr')))              driver.switch\_to.frame('mce\_0\_ifr')              content\_area = WebDriverWait(driver, wait\_max).until(                  EC.presence\_of\_element\_located((By.CSS\_SELECTOR, 'body')))  # 通常編輯器的內容區域會是 <body> 標籤              content\_area.send\_keys(content)              # 切換回主頁面              driver.switch\_to.default\_content()              # 切換到新的 iframe 並輸入 enter\_description              WebDriverWait(driver, wait\_max).until(                  EC.presence\_of\_element\_located((By.ID, 'mce\_13\_ifr')))              driver.switch\_to.frame('mce\_13\_ifr')              theinput = WebDriverWait(driver, wait\_max).until(                  EC.presence\_of\_element\_located((By.CSS\_SELECTOR, 'body')))              theinput.send\_keys(enter\_description)              # 切換回主頁面              driver.switch\_to.default\_content()              # 切換到另一個 iframe 並輸入 output\_description              WebDriverWait(driver, wait\_max).until(                  EC.presence\_of\_element\_located((By.ID, 'mce\_14\_ifr')))              driver.switch\_to.frame('mce\_14\_ifr')              theinput = WebDriverWait(driver, wait\_max).until(                  EC.presence\_of\_element\_located((By.CSS\_SELECTOR, 'body')))              theinput.send\_keys(output\_description)              # 切換回主頁面              driver.switch\_to.default\_content()              # 輸入測試數據              sampleinput = WebDriverWait(driver, wait\_max).until(                  EC.presence\_of\_element\_located((By.NAME, 'sampleinput')))              sampleinput.send\_keys(example\_input)              sampleoutput = WebDriverWait(driver, wait\_max).until(                  EC.presence\_of\_element\_located((By.NAME, 'sampleoutput')))              sampleoutput.send\_keys(example\_output)              # 點擊更新問題按鈕              update\_button = WebDriverWait(driver, wait\_max).until(                  EC.element\_to\_be\_clickable((By.NAME, 'updateProblem')))              update\_button.click()              # 處理測試資料上傳              infilepattern = WebDriverWait(driver, wait\_max).until(                  EC.presence\_of\_element\_located((By.NAME, 'infilepattern')))              infilepattern.clear()              infilepattern.send\_keys(f'{problem\_id}\_##.in')              outfilepattern = WebDriverWait(driver, wait\_max).until(                  EC.presence\_of\_element\_located((By.NAME, 'outfilepattern')))              outfilepattern.clear()              outfilepattern.send\_keys(f'{problem\_id}\_##.out')              # 找到上傳檔案的 input 元素              testdatas = WebDriverWait(driver, wait\_max).until(                  EC.presence\_of\_element\_located((By.NAME, 'testdatas')))  # 確保這是 <input type="file" multiple="multiple">              print("找到檔案上傳欄位，準備上傳多個檔案...")              base\_path = os.path.dirname(os.path.abspath(\_\_file\_\_))              print(f"Base Path: {base\_path}")  # 輸出當前文件的絕對路徑              relative\_path = os.path.join(base\_path, '.','source', 'testdata', problem\_id)              print(f"Relative Path: {relative\_path}")  # 輸出計算出的相對路徑              # 列出目錄下所有檔案的完整路徑              file\_paths = []              for root, dirs, files in os.walk(relative\_path):                  for file in files:                      file\_paths.append(os.path.join(root, file))              # 檢查是否找到檔案              if not file\_paths:                  print("未找到任何檔案，請檢查路徑！")              else:                  # 將所有檔案路徑用分號分隔起來（適用於 Windows）                  all\_files = ';'.join(file\_paths)              print(f"找到的檔案: {all\_files}")              testdatas.send\_keys(all\_files)  # 上傳檔案路徑，使用分號分隔              # 點擊更新問題按鈕              uploadFiles = WebDriverWait(driver, wait\_max).until(                  EC.element\_to\_be\_clickable((By.ID, 'uploadFiles')))              uploadFiles.click()          except Exception as e:              print(e)              return []  # 返回空列表，表示登錄失敗      except Exception as e:          print("An error occurred:", e)          traceback.print\_exc()      finally:          if driver:              driver.quit()          return None | |
| 路徑 | | crawler\submit.py |
| 用途 | | 上傳題目到Judge平台 |
| 部分程式碼 | | |
| # 引入所需的模組和套件  from datetime import datetime  import re  from flask import json  from selenium import webdriver  from selenium.webdriver.chrome.service import Service  from webdriver\_manager.chrome import ChromeDriverManager  from selenium.webdriver.common.by import By  from selenium.webdriver.support.ui import WebDriverWait  from selenium.webdriver.support.ui import Select  from selenium.webdriver.support import expected\_conditions as EC  from selenium.webdriver.common.keys import Keys  # 匯入 Keys 模組來模擬按鍵動作  import pandas as pd  from time import sleep  import os  import glob  import traceback  from utils.common import ZJ\_translated\_return\_abbreviation  def CodeAlchemy\_submit(file\_name,number,title):      # crawler setting      main\_url = 'http://123.192.165.145:8081/Login'      prefs = {'profile.default\_content\_setting\_values': {'notifications': 2}}      driver = None  # 初始化 driver 變數      results = []  # 初始化 results 變數      newResult = []  # 初始化 newResult 變數      try:          chrome\_options = webdriver.ChromeOptions()          chrome\_options.add\_experimental\_option('prefs', prefs)          chrome\_options.add\_experimental\_option('excludeSwitches', ['enable-logging'])          chrome\_options.add\_argument("disable-infobars")          os.environ["WDM\_ARCH"] = "64"          driver = webdriver.Chrome(options=chrome\_options)          driver.maximize\_window()          wait\_max = 10          submit\_program\_dict = dict()          files = glob.glob(f'./source/{file\_name}')          for file\_name in files:              \_, extension = os.path.splitext(file\_name)              extension = extension.lower()              if extension == '.py':                  language = 'python'              elif extension == '.java':                  language = 'java'              elif extension == '.c':                  language = 'c'              elif extension == '.cpp':                  language = 'cpp'              else:                  continue              with open(file\_name, 'r', encoding='utf-8') as file:                  content = file.read()                  file\_key = os.path.basename(file\_name)[7:].replace("CA-", "").split('.')[0]                  submit\_program\_dict[file\_key] = content          username = 'TestCase2025'          password = 'TestCase2025'          driver.get(main\_url)          try:              login\_area = WebDriverWait(driver, wait\_max).until(EC.presence\_of\_element\_located(                  (By.CLASS\_NAME, 'col-md-4.text-center')))              login\_area = WebDriverWait(login\_area, wait\_max).until(                  EC.presence\_of\_element\_located((By.CLASS\_NAME, 'form-horizontal')))              sleep(2)              input\_username = WebDriverWait(login\_area, wait\_max).until(                  EC.presence\_of\_element\_located((By.ID, 'account')))              input\_username.send\_keys(username)              sleep(2)              input\_password = WebDriverWait(login\_area, wait\_max).until(                  EC.presence\_of\_element\_located((By.ID, 'passwd')))              input\_password.send\_keys(password)              sleep(3)              for i in range(300):                  sleep(1)                  try:                      btn\_login = WebDriverWait(login\_area, wait\_max).until(                          EC.presence\_of\_element\_located((By.CSS\_SELECTOR, 'button.btn.btn-primary')))                      btn\_login.click()                      break                  except:                      pass              else:                  raise BaseException              sleep(5)              if driver.current\_url != 'http://123.192.165.145:8081/':                  raise BaseException              else:                   print("登入成功")          except BaseException as e:              print(e)              return []  # 返回空列表，表示登錄失敗          results = []          language\_upper = language.upper()          for prob\_id in list(submit\_program\_dict.keys()):                # 在 name="searchword" 的輸入欄位內輸入文字              search\_input = WebDriverWait(driver, 10).until(                  EC.presence\_of\_element\_located((By.NAME, 'searchword'))              )              search\_input.send\_keys(title)  # 在輸入框內輸入 "{title}"              print(f"成功輸入 {title}")               # 模擬按下 Enter 鍵              search\_input.send\_keys(Keys.RETURN)  # 模擬按下 Enter 鍵來提交表單或進行搜尋              print("成功模擬按下 Enter 鍵")              # 等待結果頁面載入完成（根據實際情況調整等待條件）              WebDriverWait(driver, 10).until(EC.presence\_of\_element\_located((By.TAG\_NAME, 'table')))  # 假設結果會以表格形式呈現              # 尋找所有的 <td> 標籤              td\_elements = driver.find\_elements(By.TAG\_NAME, 'a')              # 檢查是否有任何 <td> 標籤的文字內容為 '1'              found = False              for td in td\_elements:                  if td.text == title:                      print(f"找到 {title}")                        # 點擊該 <td>1</td> 欄位                      td.click()                      print(f"成功點擊 {title}")                      found = True                      sleep(5)                      break              if not found:                  print(f"未找到 {title}")              btn\_code = WebDriverWait(driver, wait\_max).until(                  EC.presence\_of\_element\_located((By.CLASS\_NAME, "btn.btn-success")))              btn\_code.click()              btn\_py = WebDriverWait(driver, wait\_max).until(                  EC.element\_to\_be\_clickable((By.CSS\_SELECTOR, f"input[name='language'][value='{language\_upper}']")))              btn\_py.click()              input\_code = WebDriverWait(driver, wait\_max).until(                  EC.presence\_of\_element\_located((By.ID, "code")))              input\_code.send\_keys(submit\_program\_dict[prob\_id])              btn\_submit = WebDriverWait(driver, wait\_max).until(                  EC.presence\_of\_element\_located((By.ID, "submitCode")))              btn\_submit.click()              sleep(1)              table\_result = WebDriverWait(driver, wait\_max).until(                  EC.presence\_of\_element\_located((By.CLASS\_NAME, "table.table-hover")))              current\_row = WebDriverWait(table\_result, wait\_max).until(                  EC.presence\_of\_all\_elements\_located((By.TAG\_NAME, "tr")))[1]              sleep(6)              row\_data = []              for col in WebDriverWait(current\_row, wait\_max).until(                      EC.presence\_of\_all\_elements\_located((By.TAG\_NAME, "td"))):                  row\_data.append(col.text.strip())              results.append(row\_data)          newResult = []          for i in range(len(results[0])):  # 遍歷結果的每一列              if i == 1:                  newResult.append(int(number))              elif i == 2:                  newResult.append('CA-' + results[0][i][:4])              elif i == 3:                  if results[0][i].startswith('AC'):                      message, ensue = ZJ\_translated\_return\_abbreviation(results[0][i])                      newResult.append(ensue)                      if ensue == 'Accepted':                          match = re.search(r'\((\d+ms),\s([\d.]+MB)\)', results[0][i])                          if match:                              run\_time = match.group(1)                              memory = match.group(2)                              newResult.append(run\_time)                              newResult.append(memory)                      else:                          newResult.append('0')                          newResult.append('0')              elif i == 5:                  newResult.append(results[0][i])              elif i == 4:                  newResult.append(results[0][i].lower())          if newResult:              df = pd.DataFrame([newResult])              csv\_data = df.to\_csv(index=False, header=False, encoding='utf-8').strip()              with open('./crawler/result.csv', 'a', encoding='utf-8') as f:                  f.write(csv\_data)                  f.write('\n')      except Exception as e:          print("An error occurred:", e)          traceback.print\_exc()      finally:          if driver:              driver.quit()          return newResult  def TIOJ\_submit(file\_name, number):      main\_url = 'https://tioj.ck.tp.edu.tw/users/sign\_in'      prefs = {'profile.default\_content\_setting\_values': {'notifications': 2}}      driver = None      results = []  # 初始化 results 變數      try:          s = Service(ChromeDriverManager().install())          chrome\_options = webdriver.ChromeOptions()          chrome\_options.add\_experimental\_option('prefs', prefs)          chrome\_options.add\_experimental\_option('excludeSwitches', ['enable-logging'])          chrome\_options.add\_argument("disable-infobars")          # 強制指定為 win64 版本          os.environ["WDM\_ARCH"] = "64"          driver = webdriver.Chrome()          driver.maximize\_window()          wait\_max = 10          submit\_program\_dict = dict()          files = glob.glob(f'./source/{file\_name}')            for file\_name in files:              \_, extension = os.path.splitext(file\_name)              extension = extension.lower()                if extension == '.py':                  language = 'python'              elif extension == '.java':                  language = 'java'              elif extension == '.c':                  language = 'c'              elif extension == '.cpp':                  language = 'cpp'              with open(file\_name, 'r', encoding='utf-8') as file:                  content = file.read()                  file\_key = os.path.basename(file\_name)[7:].replace("TIOJ-", "").split('.')[0]                  submit\_program\_dict[file\_key] = content          with open('./crawler/account.json', 'r') as file:              accounts = json.load(file)['account']          for acc in accounts:              if acc[0][0-len(str(number)):]==number:                  username = acc[0]                  password = acc[1]                  driver.get(main\_url)                  try:                      input\_username = WebDriverWait(driver, wait\_max).until(                          EC.presence\_of\_element\_located((By.NAME, 'user[username]')))                      input\_username.send\_keys(username)                      sleep(1)                      input\_password = WebDriverWait(driver, wait\_max).until(                          EC.presence\_of\_element\_located((By.NAME, 'user[password]')))                      input\_password.send\_keys(password)                      sleep(1)                      sign\_in = WebDriverWait(driver, wait\_max).until(                          EC.presence\_of\_element\_located((By.NAME, 'commit')))                      sign\_in.click()                      sleep(1)                      if driver.current\_url != 'https://tioj.ck.tp.edu.tw/':                          raise BaseException                    except BaseException as e:                      print(e)                      continue                    language\_dict = {                      "python": "9",                      "cpp": "12",                      "c":'7'                  }                  value = language\_dict.get(language)                  for prob\_id in list(submit\_program\_dict.keys()):                      driver.get(f'https://tioj.ck.tp.edu.tw/problems/{prob\_id}/submissions/new')                      compiler = Select(WebDriverWait(driver, wait\_max).until(                          EC.presence\_of\_element\_located((By.ID, "submission\_compiler\_id"))))                      compiler.select\_by\_value(value)                      sleep(1)                      input\_code = WebDriverWait(driver, wait\_max).until(                          EC.presence\_of\_element\_located((By.ID, "submission\_code\_content\_attributes\_code")))                      input\_code.send\_keys(submit\_program\_dict[prob\_id])                      sleep(1)                      btn\_submit = WebDriverWait(driver, wait\_max).until(                          EC.presence\_of\_element\_located((By.ID, "form-submit-button")))                      btn\_submit.click()                      sleep(1)                      result = WebDriverWait(driver, wait\_max).until(                          EC.presence\_of\_element\_located((By.ID, "verdict")))                      results.append(result.text)                      run\_time = WebDriverWait(driver, wait\_max).until(                          EC.presence\_of\_element\_located((By.ID, "td-time-0")))                      results.append(run\_time.text)                      memory = WebDriverWait(driver, wait\_max).until(                          EC.presence\_of\_element\_located((By.ID, "td-vss-0")))                        results.append(round(int(memory.text)/1024,2))                      sleep(1)          if results:              df = pd.DataFrame([results])              csv\_data = df.to\_csv(index=False, header=False, encoding='utf-8')              with open('./crawler/result.csv', 'a', encoding='utf-8') as f:                  newResult=[]                  f.write(number + ",")                  newResult.append(number)                  f.write('TIOJ-'+file\_key + ",")                  newResult.append(f'TIOJ-{file\_key}')                  f.write(csv\_data.strip() + ",")                  for i in csv\_data.strip().split(','):                      newResult.append(i)                  f.write(language+ ",")                  newResult.append(language)                  now\_time=str(datetime.now())# 轉換為字符串                  f.write(now\_time)                  newResult.append(now\_time)                  f.write('\n')      except Exception as e:          print("An error occurred:", e)          traceback.print\_exc()          if driver:              driver.quit()      finally:          if driver:              driver.quit()          print(newResult)          print('----------------------------------------------------------')          return newResult  def ZeroJudge\_submit(file\_name, number):      # crawler setting      main\_url = 'https://zerojudge.tw/Login'      prefs = {'profile.default\_content\_setting\_values': {'notifications': 2}}      driver = None  # 初始化 driver 變數      results = []  # 初始化 results 變數      newResult = []  # 初始化 newResult 變數      try:          chrome\_options = webdriver.ChromeOptions()          chrome\_options.add\_experimental\_option('prefs', prefs)          chrome\_options.add\_experimental\_option('excludeSwitches', ['enable-logging'])          chrome\_options.add\_extension('./crawler/extension/reCAPTCHA\_extension.crx')          chrome\_options.add\_extension('./crawler/extension/vpn\_extension\_Touch.crx')          chrome\_options.add\_argument("disable-infobars")          os.environ["WDM\_ARCH"] = "64"          driver = webdriver.Chrome(options=chrome\_options)          print(driver)          driver.maximize\_window()          wait\_max = 10          driver.get("chrome-extension://bihmplhobchoageeokmgbdihknkjbknd/panel/index.html")          sleep(2)          btn\_connect = WebDriverWait(driver, wait\_max).until(EC.presence\_of\_element\_located((By.ID, "ConnectionButton")))          btn\_connect.click()          sleep(2)          all\_windows = driver.window\_handles          driver.switch\_to.window(all\_windows[0])          submit\_program\_dict = dict()          files = glob.glob(f'./source/{file\_name}')          for file\_name in files:              \_, extension = os.path.splitext(file\_name)              extension = extension.lower()              if extension == '.py':                  language = 'python'              elif extension == '.java':                  language = 'java'              elif extension == '.c':                  language = 'c'              elif extension == '.cpp':                  language = 'cpp'              else:                  continue              with open(file\_name, 'r', encoding='utf-8') as file:                  content = file.read()                  file\_key = os.path.basename(file\_name)[7:].replace("ZJ-", "").split('.')[0]                  submit\_program\_dict[file\_key] = content          with open('./crawler/account.json', 'r') as file:              accounts = json.load(file)['account']          username = None          password = None          for acc in accounts:              if acc[0][0-len(str(number)):] == number:                  username = acc[0]                  password = acc[1]                  break          if not username or not password:              username = "TestCase2024"              password = "TestCase2024"          driver.get(main\_url)          try:              login\_area = WebDriverWait(driver, wait\_max).until(EC.presence\_of\_element\_located(                  (By.CLASS\_NAME, 'col-md-4.text-center')))              login\_area = WebDriverWait(login\_area, wait\_max).until(                  EC.presence\_of\_element\_located((By.CLASS\_NAME, 'form-horizontal')))              sleep(2)              input\_username = WebDriverWait(login\_area, wait\_max).until(                  EC.presence\_of\_element\_located((By.ID, 'account')))              input\_username.send\_keys(username)              sleep(2)              input\_password = WebDriverWait(login\_area, wait\_max).until(                  EC.presence\_of\_element\_located((By.ID, 'passwd')))              input\_password.send\_keys(password)              sleep(3)              for i in range(300):                  sleep(1)                  try:                      btn\_login = WebDriverWait(login\_area, wait\_max).until(                          EC.presence\_of\_element\_located((By.CSS\_SELECTOR, 'button.btn.btn-primary')))                      btn\_login.click()                      break                  except:                      pass              else:                  raise BaseException              sleep(5)              if driver.current\_url != 'https://zerojudge.tw/':                  raise BaseException          except BaseException as e:              print(e)              return []  # 返回空列表，表示登錄失敗          results = []          language\_upper = language.upper()          for prob\_id in list(submit\_program\_dict.keys()):              driver.get(f'https://zerojudge.tw/ShowProblem?problemid={prob\_id}')              btn\_code = WebDriverWait(driver, wait\_max).until(                  EC.presence\_of\_element\_located((By.CLASS\_NAME, "btn.btn-success")))              btn\_code.click()              btn\_py = WebDriverWait(driver, wait\_max).until(                  EC.element\_to\_be\_clickable((By.CSS\_SELECTOR, f"input[name='language'][value='{language\_upper}']")))              btn\_py.click()              input\_code = WebDriverWait(driver, wait\_max).until(                  EC.presence\_of\_element\_located((By.ID, "code")))              input\_code.send\_keys(submit\_program\_dict[prob\_id])              btn\_submit = WebDriverWait(driver, wait\_max).until(                  EC.presence\_of\_element\_located((By.ID, "submitCode")))              btn\_submit.click()              sleep(1)              table\_result = WebDriverWait(driver, wait\_max).until(                  EC.presence\_of\_element\_located((By.CLASS\_NAME, "table.table-hover")))              current\_row = WebDriverWait(table\_result, wait\_max).until(                  EC.presence\_of\_all\_elements\_located((By.TAG\_NAME, "tr")))[1]              sleep(6)              row\_data = []              for col in WebDriverWait(current\_row, wait\_max).until(                      EC.presence\_of\_all\_elements\_located((By.TAG\_NAME, "td"))):                  row\_data.append(col.text.strip())              results.append(row\_data)          newResult = []          for i in range(len(results[0])):  # 遍歷結果的每一列              if i == 1:                  newResult.append(int(number))              elif i == 2:                  newResult.append('ZJ-' + results[0][i][:4])              elif i == 3:                  if results[0][i].startswith('AC'):                      message, ensue = ZJ\_translated\_return\_abbreviation(results[0][i])                      newResult.append(ensue)                      if ensue == 'Accepted':                          match = re.search(r'\((\d+ms),\s([\d.]+MB)\)', results[0][i])                          if match:                              run\_time = match.group(1)                              memory = match.group(2)                              newResult.append(run\_time)                              newResult.append(memory)                      else:                          newResult.append('0')                          newResult.append('0')              elif i == 5:                  newResult.append(results[0][i])              elif i == 4:                  newResult.append(results[0][i].lower())          if newResult:              df = pd.DataFrame([newResult])              csv\_data = df.to\_csv(index=False, header=False, encoding='utf-8').strip()              with open('./crawler/result.csv', 'a', encoding='utf-8') as f:                  f.write(csv\_data)                  f.write('\n')      except Exception as e:          print("An error occurred:", e)          traceback.print\_exc()      finally:          if driver:              driver.quit()          return newResult | | |

|  |  |
| --- | --- |
| 路徑 | crawler\get\_problem.py |
| 用途 | 取得題目內容 |
| 部分程式碼 | |
| import csv  import os  import random  import re  import time  from bs4 import BeautifulSoup  import requests  from datetime import datetime  from utils import db    #取得ZeroJudge全部題目  def get\_ZJ\_All\_Problem():      # 讀取 CSV 文件中的問題編號      csv\_file\_path = './ZJ\_problem\_list.csv'      with open(csv\_file\_path, 'r', newline='', encoding='utf-8') as csvfile:          csv\_reader = csv.reader(csvfile)          for row in csv\_reader:              problem\_id = row[0]              ZJ\_get\_problem(problem\_id)              time.sleep(random.randint(10,20))  def ZJ\_get\_problem(problem\_id):      # 構建完整的 URL      url = f'https://zerojudge.tw/ShowProblem?problemid={problem\_id}'      # 發送 GET 請求      response = requests.get(url)      # 檢查是否成功獲取網頁      if response.status\_code == 200:          try:              # 使用 BeautifulSoup 解析 HTML              soup = BeautifulSoup(response.text, 'html.parser')              # 找到特定的 div 元素              problem\_content\_div = soup.find('div', {'id': 'problem\_content'})              problem\_theinput\_div = soup.find('div', {'id': 'problem\_theinput'})              problem\_theoutput\_div = soup.find('div', {'id': 'problem\_theoutput'})              pre\_elements = soup.find\_all('pre')              span\_element = soup.find('span', {'id': 'problem\_title'})              # 檢查是否找到                  if span\_element and problem\_content\_div and problem\_theinput\_div and problem\_theoutput\_div:                  # 獲取內容、輸入說明和輸出說明                  problem\_content = str(problem\_content\_div)                  problem\_theinput = str(problem\_theinput\_div)                  problem\_theoutput = str(problem\_theoutput\_div)                  problem\_title = span\_element.text.split()                  if type(problem\_title)==list:                      problem\_title = ' '.join(problem\_title)                  # 使用正規表達式將第一個<p>前面的所有文字和最後一個</p>後面的文字去除                  problem\_content = re.sub(r'^.\*?<p>', '<p>', problem\_content, 1)                  problem\_content = re.sub(r'</p>.\*?$', '</p>', problem\_content, 1)                  problem\_content = problem\_content[:-6]                  problem\_theinput = re.sub(r'^.\*?<p>', '<p>', problem\_theinput, 1)                  problem\_theinput = re.sub(r'</p>.\*?$', '</p>', problem\_theinput, 1)                  problem\_theoutput = re.sub(r'^.\*?<p>', '<p>', problem\_theoutput, 1)                  problem\_theoutput = re.sub(r'</p>.\*?$', '</p>', problem\_theoutput, 1)                    # 找到所有的 pre 元素                  pre\_contents = [str(pre\_element) for pre\_element in pre\_elements]                  # 將範例輸入和範例輸出分別存儲到兩個列表中                  example\_inputs = [pre\_contents[i] for i in range(0, len(pre\_contents), 2)]                  example\_outputs = [pre\_contents[i + 1] for i in range(0, len(pre\_contents), 2)]                  # 將範例輸入和範例輸出合併為同一個變數                  examples\_combined\_input = '|||'.join(example\_inputs)                  examples\_combined\_output = '|||'.join(example\_outputs)                  # 获取标签                  tag\_div = soup.find('span', class\_='tag')                  tag = tag\_div.text.strip() if tag\_div else 'N/A'                  # 假設soup是BeautifulSoup的一個實例，已經找到了對應的元素                  string = soup.find('span', title='解題統計').text.strip()                    # 使用正規表達式將非數字部分去除，並將剩下的數字放在清單中                  numbers\_only = re.findall(r'\d+', string)  # 找到所有的數字                  numbers = [int(num) for num in numbers\_only]  # 將字符串轉換為整數                  percentage=numbers[-1]                    # 根据百分比确定难度等级                  if percentage:                      percentage\_value = int(percentage)                      if percentage\_value >= 80:                          difficulty = 'easy'                      elif percentage\_value >= 60:                          difficulty = 'normal'                      else:                          difficulty = 'hard'                  else:                      difficulty = 'N/A'                  # 連接到 My SQL 資料庫                  conn=db.connection()                  # 創建一個游標對象                  cursor = conn.cursor()                  # 執行 SQL 插入語句                  sql\_insert = """                      INSERT INTO problem (problem\_id, title, content, enter\_description, output\_description, `example input`, `example output`, difficulty, tag, solved, submission, update\_time, collection)                      VALUES (%s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s)                  """                  # 提供預設值                  default\_difficulty = difficulty                  default\_tag = tag                  default\_solved = 0                  default\_submission = 0                  default\_update\_time = datetime.now()  # 如果不提供值，則預設為 NULL                  default\_collection=0                  # 使用整數值插入                  values = ('ZJ-' + problem\_id, problem\_title, problem\_content, problem\_theinput, problem\_theoutput, examples\_combined\_input,                            examples\_combined\_output, default\_difficulty, default\_tag, default\_solved, default\_submission, default\_update\_time,default\_collection)                  # 執行 SQL 插入                  cursor.execute(sql\_insert, values)                  # 提交事務                  conn.commit()                  print(f'已將題目編號 {problem\_id} 的資料儲存到 My SQL 資料庫中')              else:                  print('未找到指定的 div 元素')          except Exception as e:              print(f'處理題目編號 {problem\_id} 時出錯: {str(e)}')      else:          print(f'網頁請求失敗，狀態碼: {response.status\_code}')      response.close()  def ZJ\_get\_problem\_list():      # 存放所有問題編號的列表      problem\_ids = []      problem\_type={'BASIC':32,'CONTEST':23,'TOI':29,'UVA':29,'ORIGINAL':47}      for problem\_category, end\_page in problem\_type.items():          for page in range(1,end\_page+1):              url=f'https://zerojudge.tw/Problems?&tabid={problem\_category}&page={page}'              # 發送 GET 請求              response = requests.get(url)              # 檢查是否成功獲取網頁              if response.status\_code == 200:                  # 使用 BeautifulSoup 解析 HTML                  soup = BeautifulSoup(response.text, 'html.parser')                  # 找到所有符合條件的 a 元素                  a\_elements = soup.find\_all('a', href=lambda href: href and 'ShowProblem?problemid=' in href)                  # 檢查是否找到                  if a\_elements:                      # 提取每個<a>元素的href值，並只保留問題編號部分，加入到問題編號列表中                      problem\_ids.extend([a\_element['href'].split('=')[-1] for a\_element in a\_elements])                  else:                      print(f'第 {page} 頁未找到指定的<a>元素')              else:                  print(f'網頁請求失敗，狀態碼: {response.status\_code}')      # 寫入 CSV 文件      with open('ZJ\_problem\_list.csv', 'a', newline='', encoding='utf-8') as csvfile:          csv\_writer = csv.writer(csvfile)          for problem\_id in problem\_ids:              csv\_writer.writerow([problem\_id])      print('已將問題編號寫入 ZJ\_problem\_list.csv 文件')  def get\_TIOJ\_All\_Problem():      # 讀取 CSV 文件中的問題編號      csv\_file\_path = './TIOJ\_problem\_list.csv'      with open(csv\_file\_path, 'r', newline='', encoding='utf-8') as csvfile:          csv\_reader = csv.reader(csvfile)          for row in csv\_reader:              problem\_id = row[0]              TIOJ\_get\_problem(problem\_id)              time.sleep(random.randint(5,10))  def TIOJ\_get\_problem(problem\_id):      # 構建完整的 URL      problem\_id=str(problem\_id)      url = f'https://tioj.ck.tp.edu.tw/problems/{problem\_id}'      # 發送 GET 請求      response = requests.get(url)      # 檢查是否成功獲取網頁      if response.status\_code == 200:          try:              # 使用 BeautifulSoup 解析 HTML              soup = BeautifulSoup(response.text, 'html5lib')              # 找到所有符合條件的 <div class="panel-body">              panel\_bodies = soup.find\_all('div', class\_='panel-body')              if len(panel\_bodies) >= 7:                  # problem\_id 題目編號                  problem\_id = "TIOJ" +'-'+ problem\_id                    # title 標題                  problem\_title=''                  t = soup.find('h4', class\_='page-header').get\_text().strip()                  problem\_title = t[7:]                  # [2]難度等級percentage                  AC\_Ratio = panel\_bodies[2].text.strip()                  percentage = AC\_Ratio.split()[0][:2]                  # 根据百分比确定难度等级                  if percentage:                      percentage\_value = int(percentage)                      if percentage\_value >= 60:                          difficulty = 'easy'                      elif percentage\_value >= 40:                          difficulty = 'normal'                      else:                          difficulty = 'hard'                  else:                      difficulty = 'N/A'                  # [3]標籤Tag                  target\_panel\_body = panel\_bodies[3]                  # 找到其中的所有 <a> 標籤                  a\_tags = target\_panel\_body.find\_all('a')                  if a\_tags:                      # 如果存在 <a> 標籤，取出其文本並以空白隔開                      tag = " ".join([a.get\_text() for a in a\_tags]).strip()                  else:                      # 如果不存在 <a> 標籤，設置為 "N/A"                      tag = "N/A"                    # [4] 敘述 Description                  # 去掉<div class="panel-body">                  # 去掉</div>                  problem\_content = str(panel\_bodies[4]).replace('<div class="panel-body">', '').replace('</div>', '').strip()                  # 使用 BeautifulSoup 解析 HTML 內容                  problem\_content\_soup = BeautifulSoup(problem\_content, 'html.parser')                  # 檢查是否存在 <img> 元素                  img\_tags = problem\_content\_soup.find\_all('img')                  if img\_tags:                      # 假設圖片存儲目錄為 './static/imgs/problem\_image/TIOJ'                      image\_dir = './static/imgs/problem\_image/TIOJ'                        # 如果目錄不存在，則創建目錄                      if not os.path.exists(image\_dir):                          os.makedirs(image\_dir)                        # 遍歷每個 <img> 元素                      for img\_tag in img\_tags:                          # 獲取圖片的 URL                          img\_url = img\_tag['src']                            # 下載圖片                          img\_name = os.path.basename(img\_url)                          img\_path = os.path.join(image\_dir, img\_name)                            # 檢查圖片是否已經存在，如果不存在才下載                          if not os.path.exists(img\_path):                              try:                                  response = requests.get(f'https://tioj.ck.tp.edu.tw/{img\_url}')                                  with open(img\_path, 'wb') as f:                                      f.write(response.content)                                  print(f"圖片下載成功：{img\_url}")                                  # 修改 problem\_content 中的圖片路徑                                  problem\_content = problem\_content.replace(img\_url, f"./static/imgs/problem\_image/TIOJ/{img\_name}")                              except Exception as e:                                  print(f"圖片下載失敗：{img\_url}，錯誤信息：{e}")                          else:                              print(f"圖片已存在：{img\_url}")                              problem\_content = problem\_content.replace(img\_url, f"./static/imgs/problem\_image/TIOJ/{img\_name}")                  # [5][6] 輸出輸入說明                  problem\_theinput = str(panel\_bodies[5]).replace('<div class="panel-body">', '').replace('</div>', '').strip()                  problem\_theoutput = str(panel\_bodies[6]).replace('<div class="panel-body">', '').replace('</div>', '').strip()                  # [7][8] 輸入輸出範例                  examples = soup.find\_all('div', class\_='panel-body code-input copy-group-code')                  examples\_input = []                  examples\_output = []                  # 遍歷每個找到的元素，提取文本內容並判斷索引是奇數還是偶數                  for index, example in enumerate(examples):                      text\_content = str(example).replace('<div class="panel-body code-input copy-group-code" style="font-size:15px;line-height:18px;">', '').replace('</div>', '')                        # 如果索引是偶數，則將文本內容添加到 examples\_input 中                      if index % 2 == 0:                          examples\_input.append('<pre>'+text\_content +'</pre>' +"|||")                      # 如果索引是奇數，則將文本內容添加到 examples\_output 中                      else:                          examples\_output.append('<pre>'+text\_content +'</pre>' "|||")                  # 如果 examples\_input 非空，則刪除最後一個元素的末尾三個字元                  if examples\_input:                      examples\_input[-1] = examples\_input[-1][:-3]                  # 如果 examples\_output 非空，則刪除最後一個元素的末尾三個字元                  if examples\_output:                      examples\_output[-1] = examples\_output[-1][:-3]                  # 將 examples\_input 和 examples\_output 轉換為字符串，以方便後續操作                  examples\_input = ''.join(examples\_input).strip()                  examples\_output = ''.join(examples\_output).strip()                  # 連接到 My SQL 資料庫                  conn=db.connection()                  # 創建一個游標對象                  cursor = conn.cursor()                  # 執行 SQL 插入語句                  sql\_insert = """                      INSERT INTO problem (problem\_id, title, content, enter\_description, output\_description, `example input`, `example output`, difficulty, tag, solved, submission, update\_time, collection)                      VALUES (%s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s)                  """                  # 提供預設值                  default\_difficulty = difficulty                  default\_tag = tag                  default\_solved = 0                  default\_submission = 0                  default\_update\_time = datetime.now()  # 如果不提供值，則預設為 NULL                  default\_collection=0                  # 使用整數值插入                  values = (problem\_id, problem\_title, problem\_content, problem\_theinput, problem\_theoutput, examples\_input,                              examples\_output, default\_difficulty, default\_tag, default\_solved, default\_submission, default\_update\_time,default\_collection)                  print(values)                  # 執行 SQL 插入                  cursor.execute(sql\_insert, values)                  # 提交事務                  conn.commit()                  print(f'已將題目編號 {problem\_id} 的資料儲存到 My SQL 資料庫中')              else:                  print('未找到指定的 div 元素')          except Exception as e:              print(f'處理題目編號 {problem\_id} 時出錯: {str(e)}')      else:          print(f'網頁請求失敗，狀態碼: {response.status\_code}')      response.close()  def TIOJ\_get\_problem\_list():      # 存放所有問題編號的列表      problem\_ids = []      # 爬取指定範圍的頁面      end\_page=17      url\_pattern='https://tioj.ck.tp.edu.tw/problems?page='      for page in range(1, end\_page + 1):          # 構建完整的 URL          url = f'{url\_pattern}{page}'          # 發送GET請求          response = requests.get(url)          # 檢查回應狀態碼，200表示成功          if response.status\_code == 200:              # 解析HTML內容              soup = BeautifulSoup(response.text, 'html.parser')                # 找到目標表格              table = soup.find('table', class\_='table table-hover table-striped')                if table:                  # 找到tbody                  tbody = table.find('tbody')                    if tbody:                      # 找到所有的tr                      trs = tbody.find\_all('tr')                        # 迴圈處理每一個tr，取得第二個td的文字                      for tr in trs:                          # 找到第二個td                          second\_td = tr.find\_all('td')[1]  # 因為Python從0開始計數，所以第二個td的索引是1                            if second\_td:                              # 取得文字內容                              text = second\_td.get\_text(strip=True)                              problem\_ids.append(text)                          else:                              print("找不到第二個td")                  else:                      print("找不到tbody")              else:                  print("找不到目標表格")          else:              print("無法取得網頁內容，狀態碼:", response.status\_code)        # 寫入 CSV 文件      with open('TIOJ\_problem\_list.csv', 'a', newline='', encoding='utf-8') as csvfile:          csv\_writer = csv.writer(csvfile)          for problem\_id in problem\_ids:              csv\_writer.writerow([problem\_id])      print('已將問題編號寫入 TIOJ\_problem\_list.csv 文件') | |
| 路徑 | | crawler\registration.py |
| 用途 | | 自動註冊機器人帳號 |
| 部分程式碼 | | |
| # 引入所需的模組和套件  import uuid  from selenium import webdriver  from selenium.webdriver.chrome.service import Service  from webdriver\_manager.chrome import ChromeDriverManager  from selenium.webdriver.common.by import By  from selenium.webdriver.support.ui import WebDriverWait  from selenium.webdriver.support import expected\_conditions as EC  from time import sleep  import traceback  import json  def add\_account(username, password, json\_file):      # 讀取JSON檔案      with open(json\_file, 'r') as f:          data = json.load(f)      # 新增帳戶      data['account'].append([username, password])      # 寫入更新後的資料到JSON檔案中      with open(json\_file, 'w') as f:          json.dump(data, f, indent=4)  def ZeroJudge\_registration(number):      main\_url = 'https://zerojudge.tw/InsertUser'       # 禁用瀏覽器彈窗避免預設路徑載入失敗      driver = None  # 初始化 driver 變數      try:          s = Service(ChromeDriverManager().install())          chrome\_options = webdriver.ChromeOptions()          # 將擴充套件放入至Webdriver的開啟網頁內容            prefs = {'profile.default\_content\_setting\_values': {'notifications': 2}}          user\_agent="Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/124.0.0.0 Safari/537.36 Edg/124.0.0.0"          chrome\_options.add\_experimental\_option('prefs', prefs)          chrome\_options.add\_argument(f'user-agent={user\_agent}')          chrome\_options.add\_experimental\_option('excludeSwitches', ['enable-logging'])          chrome\_options.add\_extension('./crawler/extension/reCAPTCHA\_extension.crx')          chrome\_options.add\_extension('./crawler/extension/vpn\_extension\_Touch.crx')          # 隱藏『Chrome正在受到自動軟體的控制』這項資訊          chrome\_options.add\_argument("disable-infobars")          # 初始化WebDriver          driver = webdriver.Chrome(service=s, options=chrome\_options)          driver.maximize\_window()          wait\_max = 10          # 啟動擴充套件連上VPN          # 連結套件的html位置          driver.get("chrome-extension://bihmplhobchoageeokmgbdihknkjbknd/panel/index.html")          sleep(2)          # 找到點擊的位置並且點擊          btn\_connect = WebDriverWait(driver, wait\_max).until(EC.presence\_of\_element\_located((By.ID, "ConnectionButton")))          btn\_connect.click()          sleep(2)          # 選擇要切換的視窗根據索引號或視窗標題來選擇          all\_windows = driver.window\_handles          driver.switch\_to.window(all\_windows[0])          #註冊ZeroJudge帳號          driver.get(main\_url)          input\_code = WebDriverWait(driver, wait\_max).until(          EC.presence\_of\_element\_located((By.NAME, "account")))          input\_code.send\_keys(number)          user\_name=number.split('-')[0]          sleep(2)          #密碼          input\_code = WebDriverWait(driver, wait\_max).until(          EC.presence\_of\_element\_located((By.NAME, "passwd")))          input\_code.send\_keys(number)          password=number.split('-')[0]          sleep(2)          #確認密碼          input\_code = WebDriverWait(driver, wait\_max).until(          EC.presence\_of\_element\_located((By.NAME, "passwd2")))          input\_code.send\_keys(number)          sleep(2)          #公開暱稱          input\_code = WebDriverWait(driver, wait\_max).until(          EC.presence\_of\_element\_located((By.NAME, "username")))          input\_code.clear()          input\_code.send\_keys(number)          sleep(2)          #真實姓名          input\_code = WebDriverWait(driver, wait\_max).until(          EC.presence\_of\_element\_located((By.NAME, "truename")))          input\_code.send\_keys(number)          sleep(2)          #電子郵件          input\_code = WebDriverWait(driver, wait\_max).until(          EC.presence\_of\_element\_located((By.NAME, "email")))          input\_code.clear()          input\_code.send\_keys(f'{number}@gmail.com')          sleep(2)          #提交          for i in range(60):              sleep(1)              try:                  btn\_submit = WebDriverWait(driver, wait\_max).until(                      EC.presence\_of\_element\_located((By.XPATH, '/html/body/div[3]/div[2]/div[2]/form/div[10]/button')))                  btn\_submit.click()                  break              except:                  pass          sleep(5)          btn\_Okay = WebDriverWait(driver, wait\_max).until(              EC.presence\_of\_element\_located((By.XPATH, '/html/body/div[6]/div/div/div[3]/div/div/button')))          btn\_Okay.click()          driver.quit()          except Exception as e:          print("An error occurred:", e)          traceback.print\_exc()      finally:          if driver:              driver.quit()  def TIOJ\_registration(number):      main\_url = 'https://tioj.ck.tp.edu.tw/users/sign\_up'       # 禁用瀏覽器彈窗避免預設路徑載入失敗      driver = None  # 初始化 driver 變數      try:          s = Service(ChromeDriverManager().install())          chrome\_options = webdriver.ChromeOptions()          # 將擴充套件放入至Webdriver的開啟網頁內容            prefs = {'profile.default\_content\_setting\_values': {'notifications': 2}}          user\_agent="Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/124.0.0.0 Safari/537.36 Edg/124.0.0.0"          chrome\_options.add\_experimental\_option('prefs', prefs)          chrome\_options.add\_argument(f'user-agent={user\_agent}')          chrome\_options.add\_experimental\_option('excludeSwitches', ['enable-logging'])          chrome\_options.add\_extension('./crawler/extension/reCAPTCHA\_extension.crx')          chrome\_options.add\_extension('./crawler/extension/vpn\_extension\_Touch.crx')          # 隱藏『Chrome正在受到自動軟體的控制』這項資訊          chrome\_options.add\_argument("disable-infobars")          # 初始化WebDriver          driver = webdriver.Chrome(service=s, options=chrome\_options)          driver.maximize\_window()          wait\_max = 10          # 啟動擴充套件連上VPN          # 連結套件的html位置          driver.get("chrome-extension://bihmplhobchoageeokmgbdihknkjbknd/panel/index.html")          sleep(2)          # 找到點擊的位置並且點擊          btn\_connect = WebDriverWait(driver, wait\_max).until(EC.presence\_of\_element\_located((By.ID, "ConnectionButton")))          btn\_connect.click()          sleep(2)          # 選擇要切換的視窗根據索引號或視窗標題來選擇          all\_windows = driver.window\_handles          driver.switch\_to.window(all\_windows[0])          #註冊TIOJ帳號          driver.get(main\_url)          #帳號          input\_code = WebDriverWait(driver, wait\_max).until(          EC.presence\_of\_element\_located((By.ID, "user\_username")))          input\_code.send\_keys(number)          sleep(2)          #密碼          input\_code = WebDriverWait(driver, wait\_max).until(          EC.presence\_of\_element\_located((By.ID, "user\_password")))          input\_code.send\_keys(number)          sleep(2)          #確認密碼          input\_code = WebDriverWait(driver, wait\_max).until(          EC.presence\_of\_element\_located((By.ID, "user\_password\_confirmation")))          input\_code.send\_keys(number)          sleep(2)          #公開暱稱          input\_code = WebDriverWait(driver, wait\_max).until(          EC.presence\_of\_element\_located((By.ID, "user\_nickname")))          input\_code.clear()          input\_code.send\_keys(number)          sleep(2)          #真實姓名          input\_code = WebDriverWait(driver, wait\_max).until(          EC.presence\_of\_element\_located((By.ID, "user\_name")))          input\_code.send\_keys(number)          sleep(2)          #電子郵件          input\_code = WebDriverWait(driver, wait\_max).until(          EC.presence\_of\_element\_located((By.ID, "user\_email")))          input\_code.clear()          input\_code.send\_keys(f'{number}@gmail.com')          sleep(2)          #學校          input\_code = WebDriverWait(driver, wait\_max).until(          EC.presence\_of\_element\_located((By.ID, "user\_school")))          input\_code.clear()          input\_code.send\_keys(number)          sleep(2)          #畢業年          input\_code = WebDriverWait(driver, wait\_max).until(          EC.presence\_of\_element\_located((By.ID, "user\_gradyear")))          input\_code.clear()          input\_code.send\_keys(114)          sleep(2)            #提交          btn\_submit = WebDriverWait(driver, wait\_max).until(              EC.presence\_of\_element\_located((By.NAME, 'commit')))          btn\_submit.click()          sleep(5)          driver.quit()      except Exception as e:          print("An error occurred:", e)          traceback.print\_exc()      finally:          if driver:              driver.quit() | | |

第十章 測試模型

10-1 測試計畫

以使用者視角使用CodeAlchemy，測試以下功能：

1. 題目

1-1作答題目：確認程式是否能上傳。

1-2收藏題目：確認使用者是否將題目儲存至喜愛的題目。

1-3搜尋題目：確認使用者是否能透過搜尋功能來尋找想要的題目。

1. 比賽

2-1創建比賽：使用者是否可創建比賽。

2-2加入比賽：確認使用者可以透過比賽內容來決定是否參加比賽。

1. 會員

3-1新增會員：點擊”註冊”，使用者是否能新增會員資料。

3-2修改會員：點擊”個人資料”，使用者是否能修改個人基本資料。

1. 反饋

4-1新增反饋：使用者是否可以新增意見反饋。

4-2反饋結果：使用者是否可以查看反饋結果。

1. 題目管理

5-1新增題目：確認管理者可以新增題目至系統內。

5-2修改題目：確認管理者可修改系統內的題目內容。

5-3刪除題目：確認管理者可以刪除題目。

5-4查詢題目：確認讓管理者能更快速查詢到題目。

1. 反饋管理

6-1反饋結果：確認管理者是否能將結果反饋給使用者。

6-2反饋內容：確認管理者是否能收到使用者反饋的內容。

10-2 測試個案與測試結果

▼表 10-2-1-1 作答題目測試

|  |  |  |  |
| --- | --- | --- | --- |
| 功能編號 | 1-1 | 功能名稱 | 作答題目 |
| 測試流程 | 1.進入CodeAlchemy  2.登入帳號  3.選擇題目  4.填寫程式碼  5.上傳 | | |
| 預期結果 | 成功上傳 | | |
| 執行結果 | 上傳成功 | | |

▼表 10-2-1-2 收藏題目測試

|  |  |  |  |
| --- | --- | --- | --- |
| 功能編號 | 1-2 | 功能名稱 | 收藏題目 |
| 測試流程 | 1.進入CodeAlchemy  2.登入帳號  3.選擇題目  4.點擊"Like" | | |
| 預期結果 | 收藏成功 | | |
| 執行結果 | 收藏成功 | | |

▼表 10-2-1-3 搜尋題目測試

|  |  |  |  |
| --- | --- | --- | --- |
| 功能編號 | 1-3 | 功能名稱 | 搜尋題目 |
| 測試流程 | 1.進入CodeAlchemy  2.輸入搜尋題目相關訊息  3.顯示搜尋題目結果 | | |
| 預期結果 | 成功找到題目 | | |
| 執行結果 | 測試成功 | | |

▼表 10-2-2-1 創建比賽測試

|  |  |  |  |
| --- | --- | --- | --- |
| 功能編號 | 2-1 | 功能名稱 | 創建比賽 |
| 測試流程 | 1.進入CodeAlchemy  2.登入帳號  3.點擊"競賽"下拉選單，點擊"創建比賽"  4.輸入創建比賽相關資訊  5.點擊"創建比賽" | | |
| 預期結果 | 比賽創建成功 | | |
| 執行結果 | 創建成功 | | |

▼表 10-2-2-2 加入比賽測試

|  |  |  |  |
| --- | --- | --- | --- |
| 功能編號 | 2-2 | 功能名稱 | 加入比賽 |
| 測試流程 | 1.進入CodeAlchemy  2.登入帳號  3.點擊"競賽"下拉選單，點擊"加入比賽"  4.選擇需要加入比賽  5.點擊"加入比賽" | | |
| 預期結果 | 成功加入比賽 | | |
| 執行結果 | 測試成功 | | |

▼表 10-2-3-1 新增會員測試

|  |  |  |  |
| --- | --- | --- | --- |
| 功能編號 | 3-1 | 功能名稱 | 新增會員 |
| 測試流程 | 1.進入CodeAlchemy  2.登入帳號  3.輸入電子郵件  4.輸入註冊資料  5.點擊"註冊" | | |
| 預期結果 | 註冊成功 | | |
| 執行結果 | 註冊成功 | | |

▼表 10-2-3-2 修改會員測試

|  |  |  |  |
| --- | --- | --- | --- |
| 功能編號 | 3-2 | 功能名稱 | 修改會員 |
| 測試流程 | 1.進入CodeAlchemy  2.登入帳號  3.點擊"使用者名稱"下拉選單，點擊"個人檔案"  4.選擇需修改資料  5.點擊"儲存" | | |
| 預期結果 | 修改會員成功 | | |
| 執行結果 | 測試成功 | | |

▼表 10-2-4-1 新增反饋測試

|  |  |  |  |
| --- | --- | --- | --- |
| 功能編號 | 4-1 | 功能名稱 | 新增反饋 |
| 測試流程 | 1.進入CodeAlchemy  2.登入帳號  3.點擊"使用者名稱"下拉選單，點擊"意見反饋"  4.輸入反饋內容  5.點擊"提交反饋" | | |
| 預期結果 | 提交成功 | | |
| 執行結果 | 送出成功 | | |

▼表 10-2-4-2 反饋結果測試

|  |  |  |  |
| --- | --- | --- | --- |
| 功能編號 | 4-2 | 功能名稱 | 反饋結果 |
| 測試流程 | 1.進入CodeAlchemy  2.登入帳號  3.點擊"使用者名稱"下拉選單，點擊"意見反饋"  4.點擊"反饋紀錄" | | |
| 預期結果 | 成功查看反饋結果 | | |
| 執行結果 | 測試成功 | | |

▼表 10-2-5-1 新增題目測試

|  |  |  |  |
| --- | --- | --- | --- |
| 功能編號 | 5-1 | 功能名稱 | 新增題目 |
| 測試流程 | 1.進入CodeAlchemy  2.登入帳號  3.點擊"管理者名稱"下拉選單，點擊"新增題目"  4.輸入題目資料  5.點擊"儲存" | | |
| 預期結果 | 新增成功 | | |
| 執行結果 | 測試成功 | | |

▼表 10-2-5-2 修改題目測試

|  |  |  |  |
| --- | --- | --- | --- |
| 功能編號 | 5-2 | 功能名稱 | 修改題目 |
| 測試流程 | 1.進入CodeAlchemy  2.登入帳號  3.選擇要修改的題目，點擊"筆"的圖示  4.修改題目內容  5.點擊"儲存" | | |
| 預期結果 | 修改成功 | | |
| 執行結果 | 測試成功 | | |

▼表 10-2-5-3 刪除題目測試

|  |  |  |  |
| --- | --- | --- | --- |
| 功能編號 | 5-3 | 功能名稱 | 刪除題目 |
| 測試流程 | 1.進入CodeAlchemy  2.登入帳號  3.選擇要刪除的題目  4.點擊"垃圾桶"圖示 | | |
| 預期結果 | 刪除成功 | | |
| 執行結果 | 測試成功 | | |

▼表 10-2-5-4 查詢題目測試

|  |  |  |  |
| --- | --- | --- | --- |
| 功能編號 | 5-4 | 功能名稱 | 查詢題目 |
| 測試流程 | 1.進入CodeAlchemy  2.輸入查詢題目相關訊息  3.顯示查詢題目結果 | | |
| 預期結果 | 查詢成功 | | |
| 執行結果 | 查詢成功 | | |

▼表 10-2-6-4 反饋結果測試

|  |  |  |  |
| --- | --- | --- | --- |
| 功能編號 | 6-1 | 功能名稱 | 反饋結果 |
| 測試流程 | 1.進入CodeAlchemy  2.登入帳號  3.點擊"管理者名稱"下拉選單，點擊"意見反饋" | | |
| 預期結果 | 成功查看反饋結果 | | |
| 執行結果 | 測試成功 | | |

▼表 10-2-6-2 反饋內容測試

|  |  |  |  |
| --- | --- | --- | --- |
| 功能編號 | 6-2 | 功能名稱 | 反饋內容 |
| 測試流程 | 1.進入CodeAlchemy  2.登入帳號  3.點擊"管理者名稱"下拉選單，點擊"意見反饋"  4.選擇要回覆的反饋，點擊"回覆"  5.輸入反饋內容  6.點擊"回覆" | | |
| 預期結果 | 回覆成功 | | |
| 執行結果 | 送出成功 | | |

第十一章 操作手冊

▼表 11-1 操作手冊

|  |  |
| --- | --- |
| 系統元件資訊 | |
| 元件名稱 | CodeAlchemy |
| 版本 | V1.0 |
| 系統類別 | 程式比對 |
| 軟體類別 | 數據分析 |
| 支援語言 | 繁體中文 |
| 最低版本需求 | macOS13.1.1  Window10 |
| 價格 | 免費 |
| 需要權限 | 基本個人資料  網路存取權 |

第十二章 使用手冊

進入CodeAlchemy

一張含有 文字, 螢幕擷取畫面, 軟體, 數字 的圖片

自動產生的描述

使用者登入

一張含有 文字, 螢幕擷取畫面, 軟體, 電腦圖示 的圖片

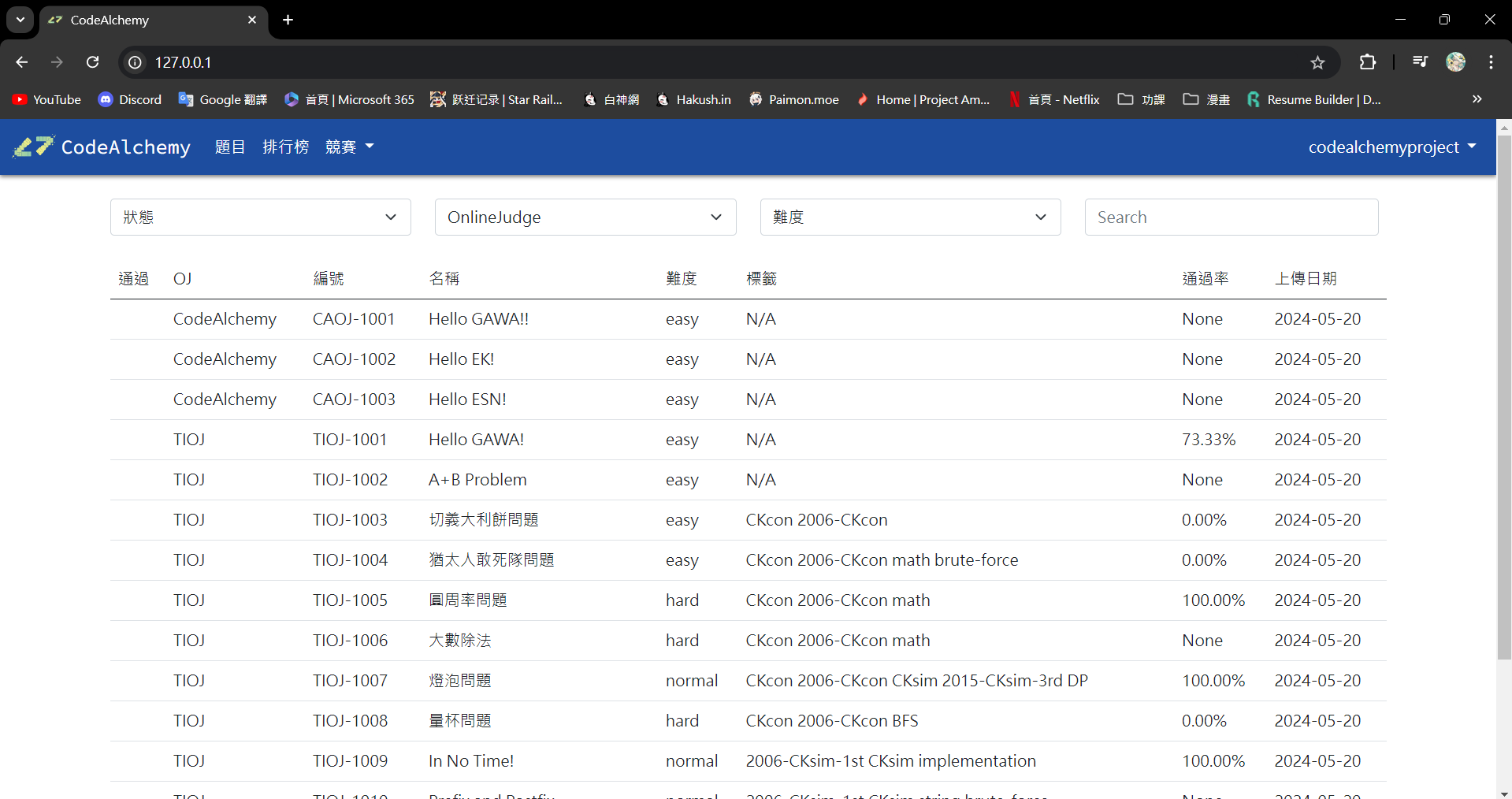
自動產生的描述

使用者註冊

一張含有 文字, 螢幕擷取畫面, 軟體, 電腦圖示 的圖片

自動產生的描述

使用者查看題目列表



使用者上傳題目

一張含有 文字, 螢幕擷取畫面, 軟體, 電腦圖示 的圖片

自動產生的描述

點擊"程式碼分析"

一張含有 文字, 軟體, 電腦圖示, 網頁 的圖片

自動產生的描述

點擊"作答情況"

一張含有 螢幕擷取畫面, 文字, 軟體, 電腦圖示 的圖片

自動產生的描述

使用者查看排行榜

一張含有 文字, 螢幕擷取畫面, 軟體, 電腦圖示 的圖片

自動產生的描述

使用者創建比賽

一張含有 文字, 螢幕擷取畫面, 軟體, 電腦圖示 的圖片

自動產生的描述

點選"新增題目"列表

一張含有 文字, 軟體, 電腦圖示, 網頁 的圖片

自動產生的描述

使用者加入比賽

一張含有 文字, 螢幕擷取畫面, 軟體, 電腦圖示 的圖片

自動產生的描述

點選"加入比賽"後查看題目

一張含有 螢幕擷取畫面, 文字, 軟體, 電腦圖示 的圖片

自動產生的描述

點選"加入比賽"後查看排名

一張含有 文字, 螢幕擷取畫面, 軟體, 電腦圖示 的圖片

自動產生的描述

使用者個人檔案

一張含有 文字, 螢幕擷取畫面, 軟體, 電腦圖示 的圖片

自動產生的描述

使用者個人解題紀錄

一張含有 文字, 螢幕擷取畫面, 軟體, 電腦圖示 的圖片

自動產生的描述

使用者收藏紀錄

一張含有 螢幕擷取畫面, 軟體, 文字, 電腦圖示 的圖片

自動產生的描述

使用者反饋

一張含有 文字, 螢幕擷取畫面, 軟體, 電腦圖示 的圖片

自動產生的描述

點選"反饋紀錄"

一張含有 螢幕擷取畫面, 文字, 軟體, 電腦圖示 的圖片

自動產生的描述

管理者題目列表

一張含有 文字, 螢幕擷取畫面, 軟體, 電腦圖示 的圖片

自動產生的描述

管理者修改題目

一張含有 文字, 螢幕擷取畫面, 軟體, 電腦圖示 的圖片

自動產生的描述

管理者新增題目

一張含有 文字, 螢幕擷取畫面, 軟體, 電腦圖示 的圖片

自動產生的描述

管理意見反饋

一張含有 文字, 螢幕擷取畫面, 軟體, 電腦圖示 的圖片

自動產生的描述

點擊"回覆"

一張含有 螢幕擷取畫面, 文字, 軟體, 電腦圖示 的圖片

自動產生的描述

第十三章 心得感想

10956010 童柏祐

在這次畢業專題的製作過程中，負責後端與爬蟲的部分是相當具有挑戰性且充滿學習機會的。以下是我對於這段經歷的一些心得感想：

1.技術整合的挑戰

後端開發與爬蟲技術的結合，讓我體會到不同技術之間的相互依賴。後端不僅要處理資料庫的交互，也需要有效地管理爬蟲所獲取的數據。這要求我在設計 API 時，必須充分考慮到數據的格式與結構，以便能夠順利地將爬取的數據存入資料庫，並且能夠透過前端輕鬆地呈現給用戶。

2.爬蟲實作的問題

在實作爬蟲的過程中，起初我並沒有太多經驗，這讓我遇到了一些挑戰。特別是因為連線數量過多，我的爬蟲曾被 Zerojudge 封鎖，這對我來說是一個很大的挫折。面對這樣的困境，我開始尋找解決方案。

為了避免再次被封鎖，我開始使用 VPN 來調整連線數量，這樣可以有效地分散請求，減少被系統偵測到的風險。此外，面對網站的反爬蟲措施，例如 Google 的 reCAPTCHA，我也運用了 AI 技術來協助處理。這些策略不僅提高了我爬蟲的穩定性，也讓我對爬蟲技術有了更深入的理解，增強了我的問題解決能力。這次的經驗讓我明白了在爬蟲開發中，效能優化與反制措施的重要性，也讓我在未來的專案中能夠更加靈活地應對各種挑戰。

3.團隊合作與溝通

在專題中，與其他成員的合作也讓我深刻體會到溝通的重要性。後端開發的進展需要與前端緊密結合，這要求我定期與周恩宇討論 API 的設計與數據結構，以確保雙方在專案上的步調一致。這不僅促進了我的溝通能力，也增強了團隊合作的默契。

總結

這次畢業專題的製作過程不僅讓我學到了許多技術知識，還提升了我的問題解決能力與團隊合作能力。面對挑戰時，我學會了靈活應對與持續學習，這些經歷將會在我未來的職業生涯中發揮重要作用。希望未來有更多的機會來探索這些技術的深度與廣度！

10956011 周恩宇

10956030 林妤玟

10956044 顏子綺

10956051 范雨茜

第十四章 參考資料

附錄一 評審建議與回應

初評

|  |  |
| --- | --- |
| 評審建議事項 | 修改情形 |
| 強調系統可透過Dolos分析上傳程式是否抄襲，是否具有意義（若為作業抄襲問題偵測，可行，但於競賽時解題查核，則意義不大），解題程式寫作受限於程式語法，相似性極高。建議可強調Dolos 其他功能整合到系統的功能，或將使用時機談清楚。 |  |
| 解題程式語言是透過什麼核心服務執行程式分析、結果驗證及整合外部哪些系統服務，未見於系統架構圖表示，請補上。 |  |
| 系統希望能開設新學習課程功能，但未針對此部份進一步說明內容為何？如何實現於系統中，且於資料表亦未見到此部份規劃，請重新增補或修正。 | 已取消這部分功能 |
| T02資料表欄位有問題，請修正。 | 已修正 |
| T04 是否需要為Problem 補上類別等欄位，便於分類，請研議。 |  |
| 使用現成套件應說明清楚 |  |
| 提供分析實例 |  |
| 系統目標應更明確聚焦 |  |
| 主題與目標不明確，有點雜 |  |
| 商業模式與內容不吻合 | 已明確改為免費，不再是收費 |
| 程式碼抄襲方向，與題目目標相關性很低 |  |
| 建議重新勾勒題目與內容 |  |