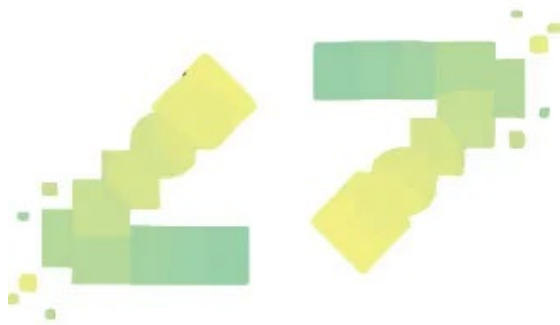


國立臺北商業大學

資訊管理系

113' 資訊系統專案設計

系統手冊



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 113 年 10 月 14 日

目錄

第一章 前言	1
1-1 背景簡介	1
1-2 動機	1
1-3 系統目的與目標	2
1-4 預期成果	2
第二章 營運計畫	3
2-1 可行性分析	3
2-2 商業模式	3
2-3 市場分析-STP	3
2-4 競爭力分析 SWOT-TOWS	4
第三章 系統規格	5
3-1 系統架構	5
3-2 系統開發工具	5
3-3 系統使用環境開發工具	6
第四章 專案時程與組織分工	7
4-1 專案時程圖	7
4-2 專案組織與分工	8

第五章 需求模型	12
5-2 需求清單	13
第六章 程序模型	14
6-1 資料流程圖(Data flow diagram).....	14
6-2 程序規格書(Process specification).....	19
第七章 資料模型	24
7-1 實體關聯圖(Entity relationship diagram)	24
7-2 資料字典(Data dictionary)	25
第八章 資料庫設計	26
8-1 資料庫關聯表(Constraints).....	26
8-2 表格及其 Meta data	27
第九章 程式	31
9-1 軟體架構與程式清單.....	31
9-2 程式規格描述.....	33
第十章 測試模型	134
10-1 測試計畫.....	134
10-2 測試個案與測試結果.....	135
第十一章 操作手冊	141

第十二章 使用手冊	142
第十三章 心得感想	155
第十四章 參考資料	158
附錄一 評審建議與回應	159

圖目錄

圖 3-1-1 系統架構	5
圖 4-1-1 專案時程	7
圖 5-1-1 功能分解圖	12
圖 6-1-1 系統環境圖	14
圖 6-1-2 DFD 圖 0	15
圖 6-1-3 DFD 圖 1	16
圖 6-1-4 DFD 圖 2	16
圖 6-1-5 DFD 圖 3	17
圖 6-1-6 DFD 圖 4	17
圖 6-1-7 DFD 圖 5	18
圖 6-1-8 DFD 圖 6	18
圖 7-1-1 實體關聯圖	24
圖 8-1-1 資料庫關聯圖	26

表目錄

表 2-4-1 SWOT 分析	4
表 2-4-2 TOWS 分析	4
表 3-2-1 系統軟、硬體需求	5
表 3-3-1 開發工具	6
表 4-2-1 專案組織分工表	8
表 4-2-2 專題成果工作內容與貢獻度表	10
表 5-2-1 功能性需求	13
表 5-2-2 非功能性需求	13
表 6-2-1 程序規格書-題目作答	19
表 6-2-2 程序規格書-收藏題目	19
表 6-2-3 程序規格書-搜尋題目	19
表 6-2-4 程序規格書-創建比賽	20
表 6-2-5 程序規格書-加入比賽	20
表 6-2-6 程序規格書-新增會員	20
表 6-2-7 程序規格書-會員資料	21
表 6-2-8 程序規格書-新增回饋	21
表 6-2-9 程序規格書-回饋結果	21
表 6-2-10 程序規格書-新增題目	22
表 6-2-11 程序規格書-修改題目	22

表 6-2-12 程序規格書-刪除題目	22
表 6-2-13 程序規格書-查詢題目	23
表 6-2-14 程序規格書-回饋結果	23
表 6-2-15 程序規格書-回饋內容	23
表 6-2-16 程序規格書-使用者管理	23
表 7-2-1 資料字典	25
表 8-2-1 user 資料表	27
表 8-2-2 feedback 資料表	27
表 8-2-3 problem 資料表	28
表 8-2-4 answer record 資料表	29
表 8-2-5 collection 資料表	29
表 8-2-6 contest problem 資料表	29
表 8-2-7 contest 資料表	30
表 8-2-8 contest participant 資料表	30
表 8-2-9 contest submission 資料表	30
表 9-1-1 前端程式清單	31
表 9-1-2 後端程式清單	32
表 9-2-1 程式清單	33
表 10-2-1-1 作答題目測試	135
表 10-2-1-2 收藏題目測試	135

表 10-2-1-3 搜尋題目測試	135
表 10-2-2-1 創建比賽測試	136
表 10-2-2-2 加入比賽測試	136
表 10-2-3-1 新增會員測試	136
表 10-2-3-2 修改會員測試	137
表 10-2-4-1 新增回饋測試	137
表 10-2-4-2 回饋結果測試	137
表 10-2-5-1 新增題目測試	138
表 10-2-5-2 修改題目測試	138
表 10-2-5-3 刪除題目測試	138
表 10-2-5-4 查詢題目測試	139
表 10-2-6-1 回饋結果測試	139
表 10-2-6-2 回饋內容測試	139
表 11-1 操作手冊	141
表 12-1 進入 CodeAlchemy	142
表 12-2 使用者登入	142
表 12-3 使用者註冊	143
表 12-4 使用者查看題目列表	143
表 12-5 使用者上傳題目	144
表 12-6 程式碼分析	144

表 12-7 作答情況.....	145
表 12-8 使用者查看排行榜.....	145
表 12-9 使用者創建比賽.....	146
表 12-10 新增題目	146
表 12-11 使用者加入比賽.....	147
表 12-12 私人比賽.....	147
表 12-13 比賽題目	148
表 12-14 比賽排名	148
表 12-15 使用者個人檔案.....	149
表 12-16 使用者個人解題紀錄.....	149
表 12-17 使用者收藏紀錄.....	150
表 12-18 使用者回饋.....	150
表 12-19 回饋紀錄.....	151
表 12-20 管理者題目列表.....	151
表 12-21 管理者修改題目	152
表 12-22 管理者新增題目	152
表 12-23 管理意見回饋.....	153
表 12-24 回饋回覆.....	153

第一章 前言

1-1 背景簡介

隨著數位時代的快速發展，程式設計已成為現代社會不可或缺的一環。對學生、工程師和技術人員來說，具備紮實的程式設計基礎和解決問題的能力，已經成為進入高科技產業的關鍵要素之一。

在過去的學習過程中，我們花費了大量時間尋找高品質的學習資源。雖然現有的學習平台提供了豐富的題目，讓使用者可以自主練習。但我們發現傳統的教育方式缺乏趣味性和互動性，難以真正激發學習者的興趣和持續投入。

針對這些問題，我們希望開發一個類似 ZeroJudge 的學習平台，提供線上解題功能，讓使用者可以通過實際操作來鞏固程式設計的知識。我們的平台包含大量多樣化的題目，涵蓋各種程式設計基礎與進階主題，幫助學習者在循序漸進的過程中提高自己的解題能力。

我們的平台還將引入 Dolos 系統，Dolos 是一款先進的程式碼比對工具，幫助學習者了解不同的解題方法，並理解更優秀的解題思路。我們希望通過結合 ZeroJudge 的豐富題庫功能和 Dolos 系統的分析能力，為學習者打造一個強大而有效的學習工具，提升其程式設計能力、實踐經驗，並助其應對未來挑戰。

1-2 動機

我們的動機源於對教育貢獻的渴望，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...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. Dolos 解法分群：我們將 Dolos 與平台進行深度整合，這項技術可以快速分析並將使用者送出的程式碼根據不同的程式結構、演算法以及語法特徵進行分類。平台將利用這些分類結果，通過視覺化的方式展示同一題目下的多種解法，幫助學習者在不同的解法之間進行比較和分析。

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 Copylocks 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：CodeAlchemy 的目標是為開發人員、學生及技術專業人士提供一個強大的程式學習與效能提升平台。同時，CodeAlchemy 結合 Dolos 系統，能夠快速對程式碼進行分群，讓使用者快速了解不同解法，幫助學習者更好地理解程式設計思路，從而提高程式碼的品質與創造力。
2. 教育與學術市場模式：CodeAlchemy 將 Dolos 系統應用於學術市場，為教育機構提供強大的程式碼分群與分析功能，幫助教師和學生快速了解同一題目的多種解法。透過分析程式碼結構與效率，教師能夠更精確地評估學生的學習進展，並根據解法提供具體的回饋。這不僅幫助學生增強解題能力，還能激發學習興趣，讓程式設計過程更加互動、有趣。

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

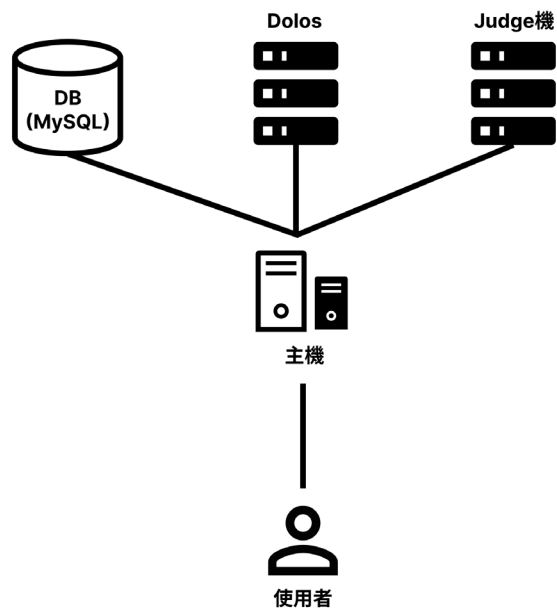
優勢(Strengths)	弱勢(Weaknesses)
<ol style="list-style-type: none"> 1. 提供中文題庫，讓看不懂英文的人也能輕易讀懂題目。 2. 提供了題目解題影片，當使用者解題時對題目毫無頭緒時可以從解題影片中尋求解法。 3. 利用 Dolos 系統了解程式中的相似性，讓使用者反思寫程式時的習慣，從而提升程式的創新性。 	<ol style="list-style-type: none"> 1. 要持續開發和維護一個高品質的教育平台需要大量的技術和資源。 2. 無法及時調整和改進平台，可能會影響使用者的滿意度。 3. 系統無法處理某些程式語言，需要進一步改善。
機會(Opportunities)	威脅(Threats)
<ol style="list-style-type: none"> 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。 	<ol style="list-style-type: none"> 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



▲ 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，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 22GB 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、CSS、JavaScript
前端套件	 Bootstrap  	Bootstrap、CodeMirror 5、Dolos
後端	 	Python、Flask
後端工具	 	Docker、Oracle Virtualbox
資料庫		MySQL
介面及插圖繪製工具		
插圖	 	Bootstrap Icons、Adobe illustrator
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

■ 預期進度
■ 實際進度

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		●			
	回饋管理					●
	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		●			
	回饋					●

項目/組員		10956010 童柏祐	10956011 周恩宇	10956030 林妤玟	10956044 顏子綺	10956051 范雨茜
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 <u>童柏祐</u>	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、送出使用者程式碼並顯示結果、處理作答紀錄、爬蟲。	<u>20</u> %
2	組員 <u>周恩宇</u>	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。	<u>20</u> %
3	組員 <u>林好玟</u>	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、測試模型、操作手冊、使用手冊。	<u>20</u> %
4	組員 <u>顏子綺</u>	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。	<u>20</u> %
5	組員 <u>范雨茜</u>	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，回饋與回饋管理的前後端程式與頁面撰寫。	<u>20</u> %
			總計:100%

10956010 童柏祐

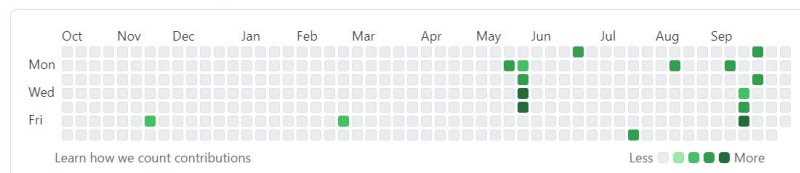



BoYoOuO

Popular repositories

BoYoOuO doesn't have any public repositories yet.

31 contributions in the last year





chouenyu0808

Follow

Popular repositories

flask-app-1124Public

HTML

123 contributions in the last year

	Oct	Nov	Dec	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep
Mon												
Wed												
Fri												

Learn how we count contributions

LessMore



fishball0612

Follow

Pinned

flask-app-1124Public

HTML

31 contributions in the last year

	Oct	Nov	Dec	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep
Mon												
Wed												
Fri												

Learn how we count contributions

LessMore



10956044

Follow

Popular repositories

flask-app-1124Public

HTML

CodeAlchemy

Forked from CodeAlchemyproject/CodeAlchemy


JavaScript

70 contributions in the last year

	Oct	Nov	Dec	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct
Mon													
Wed													
Fri													

Learn how we count contributions

LessMore



fanyucian

Follow

Popular repositories

flask-app-1124Public

HTML

-

52 contributions in the last year

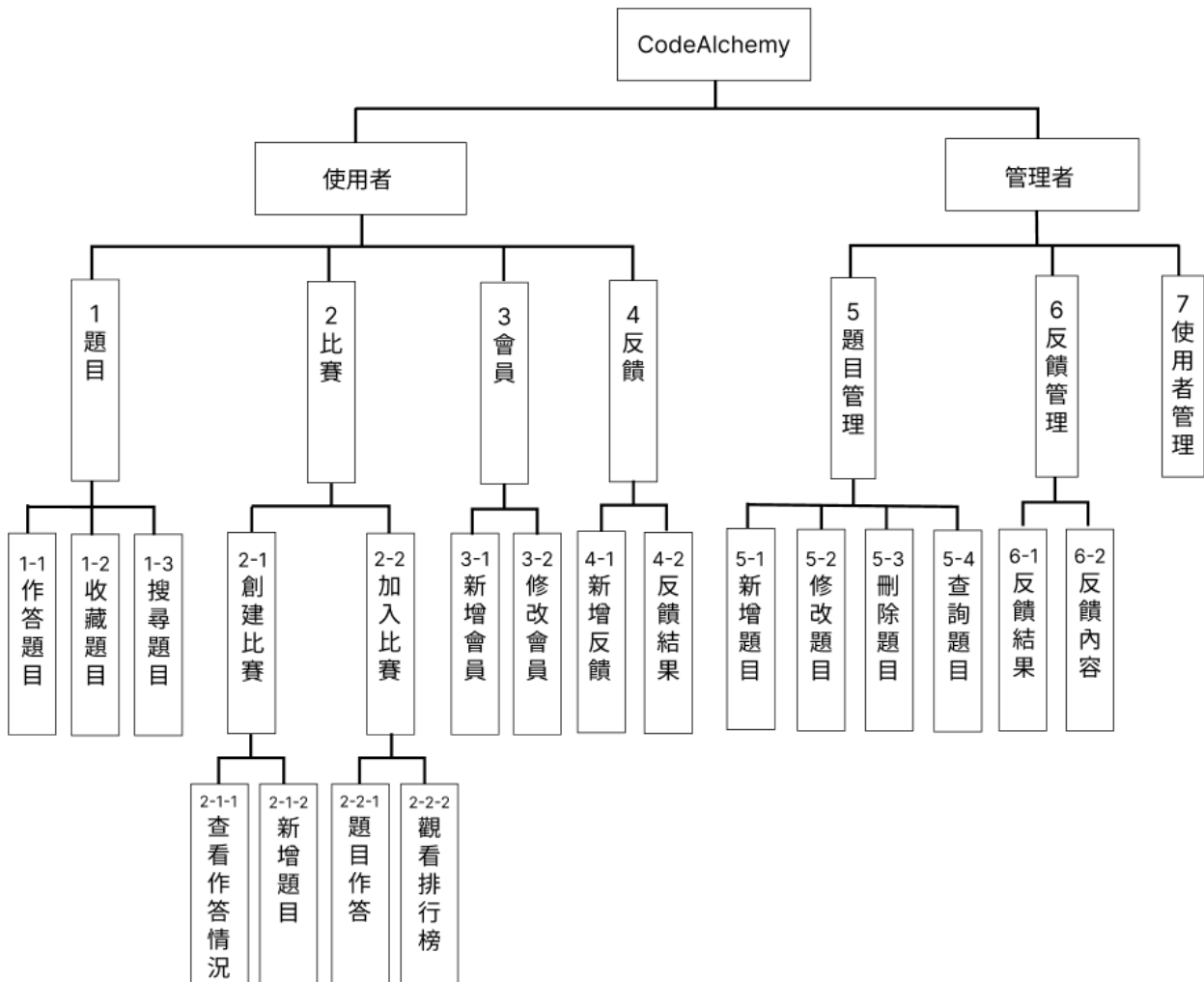
	Oct	Nov	Dec	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep
Mon												
Wed												
Fri												

Learn how we count contributions

LessMore

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

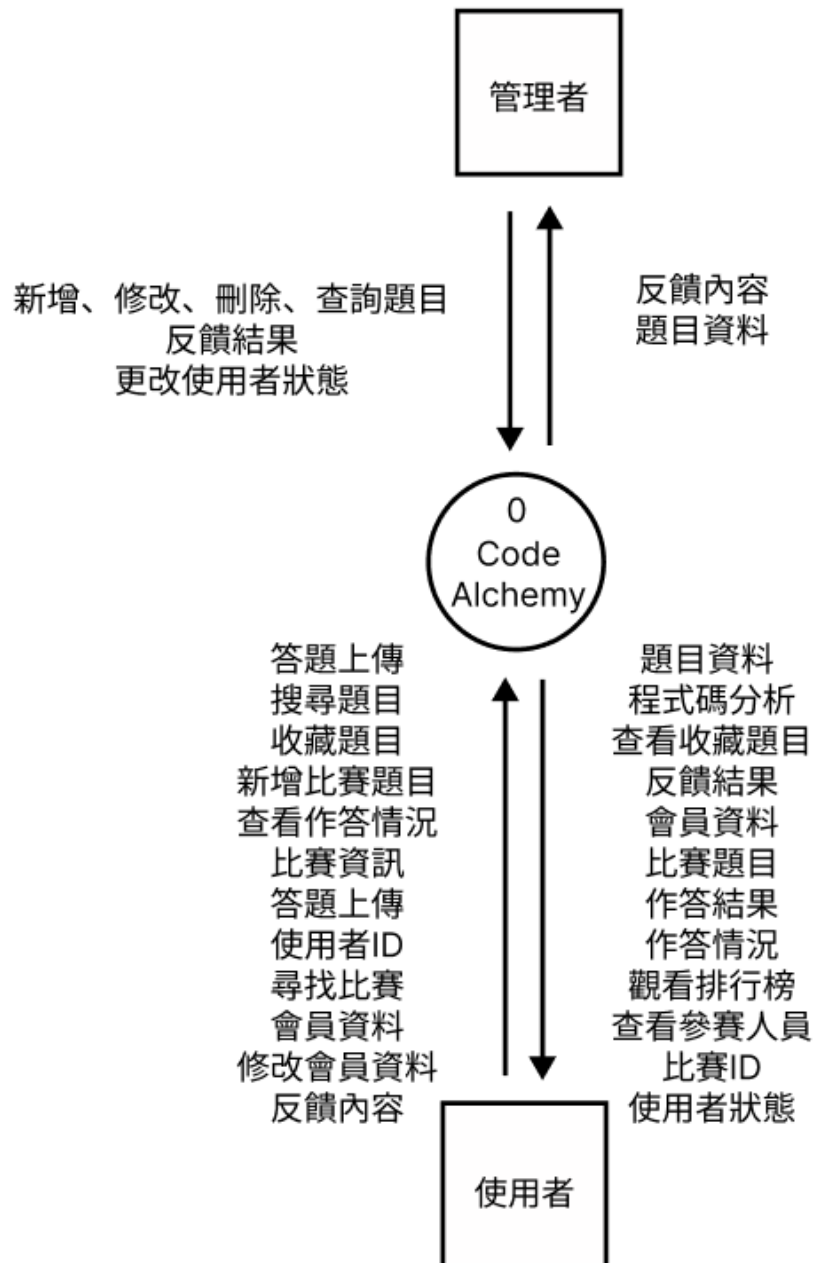
編號	功能	功能敘述
1	題目	1-1 作答題目：點擊題目，輸入程式即可上傳。 1-2 收藏題目：為使用者儲存喜愛的題目。 1-3 搜尋題目：使用者可透過搜尋功能來尋找想要的題目。
2	比賽	2-1 創建比賽：使用者可創建比賽來考驗使用者的程式語言能力。 2-2 加入比賽：可以觀看比賽內容來決定是否參加比賽。
3	會員	3-1 新增會員：點擊”註冊”，使用者可以新增會員資料到系統內。 3-2 修改會員：點擊”個人資料”即可查看使用者存到系統內的個人基本資料。
4	回饋	提供使用者來對系統意見回饋。
5	題目管理	5-1 新增題目：管理者可以新增題目至系統內。 5-2 修改題目：管理者可修改系統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

▼表 5-2-2 非功能性需求

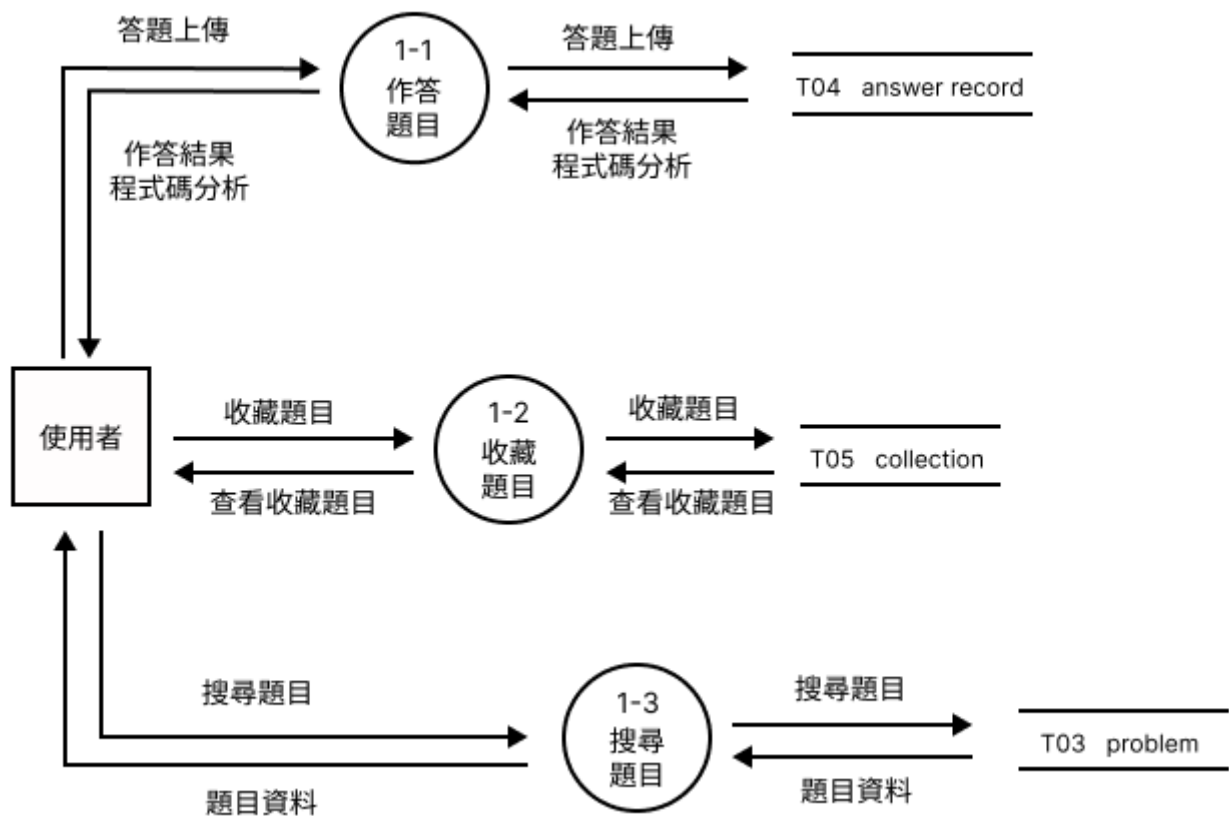
功能	功能敘述
效能需求	確保系統能夠快速、穩定地處理大量的用戶請求和數據操作。最大限度地最佳化系統性能，盡量減少頁面載入時間和服務器延遲。
可用性	提供直覺友好的用戶界面，讓使用者能夠輕鬆找到所需要的功能並進行操作。
穩定性	測試系統的穩定性和可靠性，確保在高流量和高負載情況下仍能正常運行。

第六章 程序模型

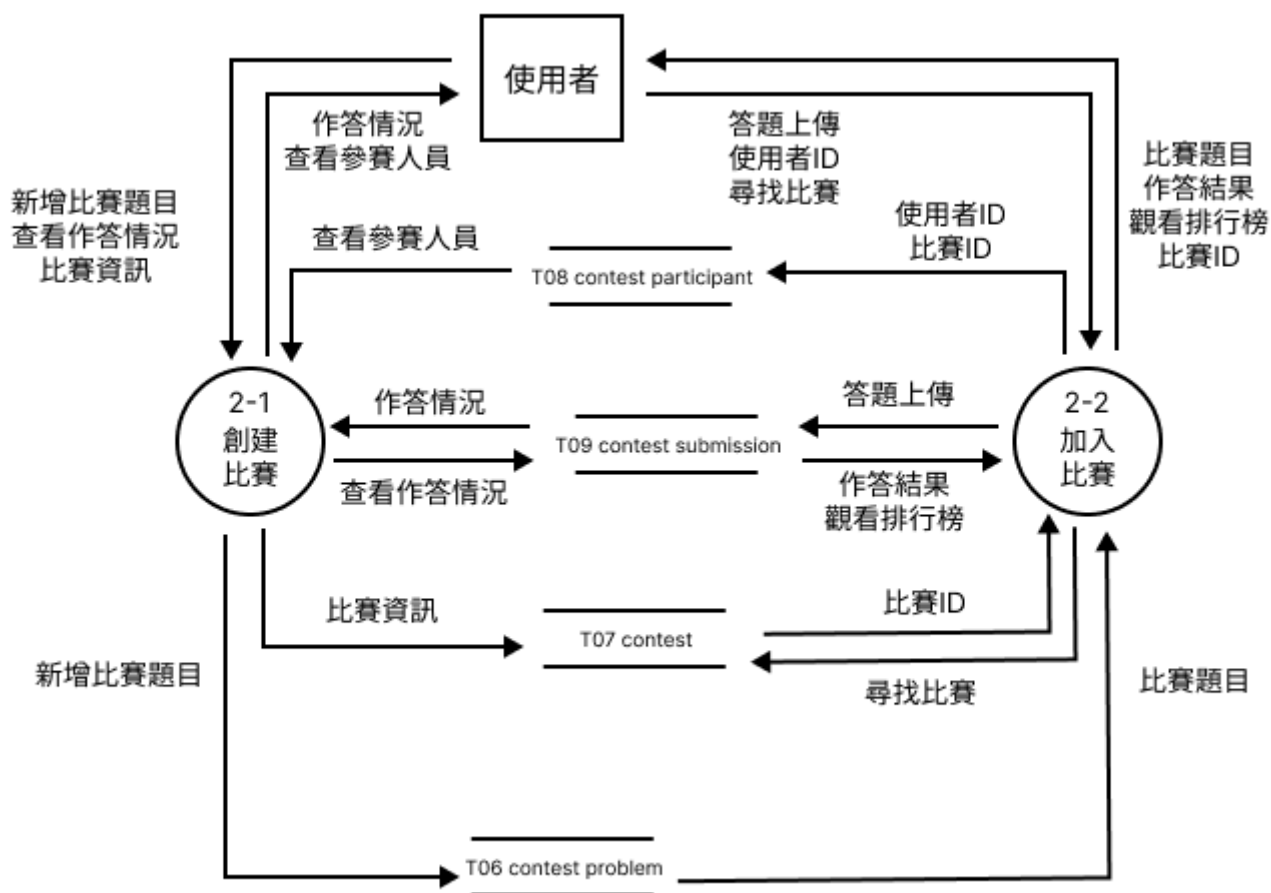
6-1 資料流程圖(Data flow diagram)



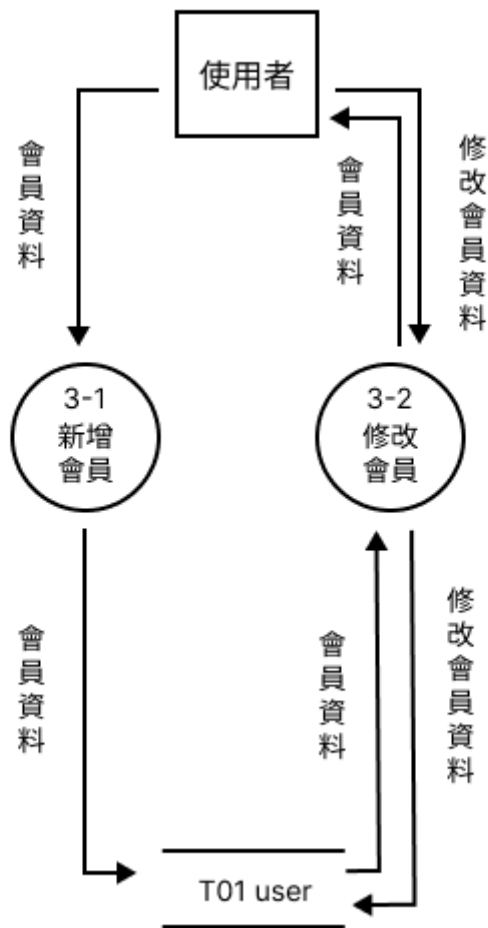
▲ 圖 6-1-1 系統環境圖



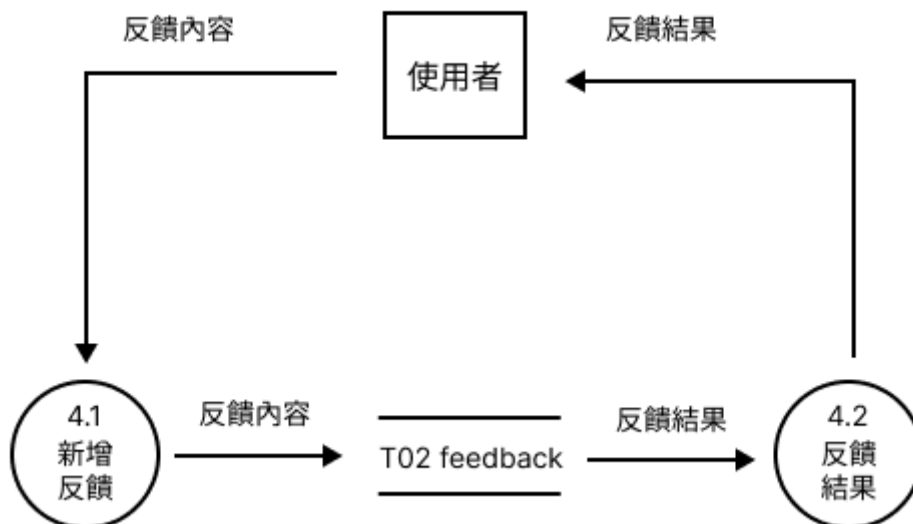
▲ 圖 2-1-3 DFD 圖 1



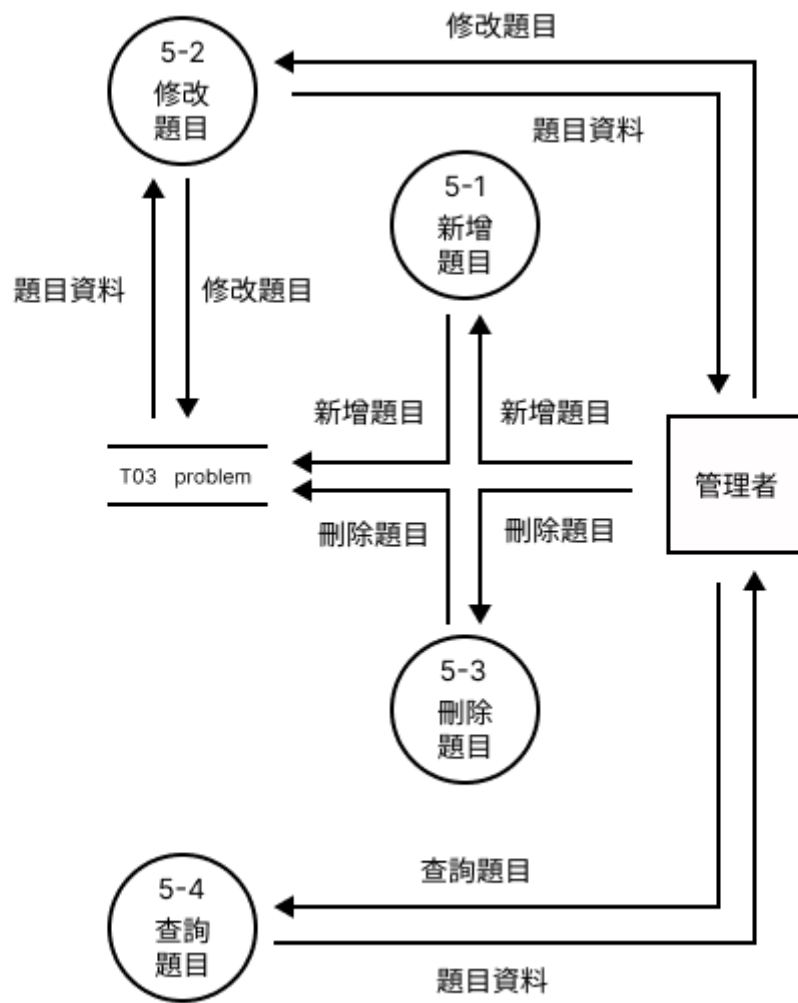
▲ 圖 6-1-4 DFD 圖 2



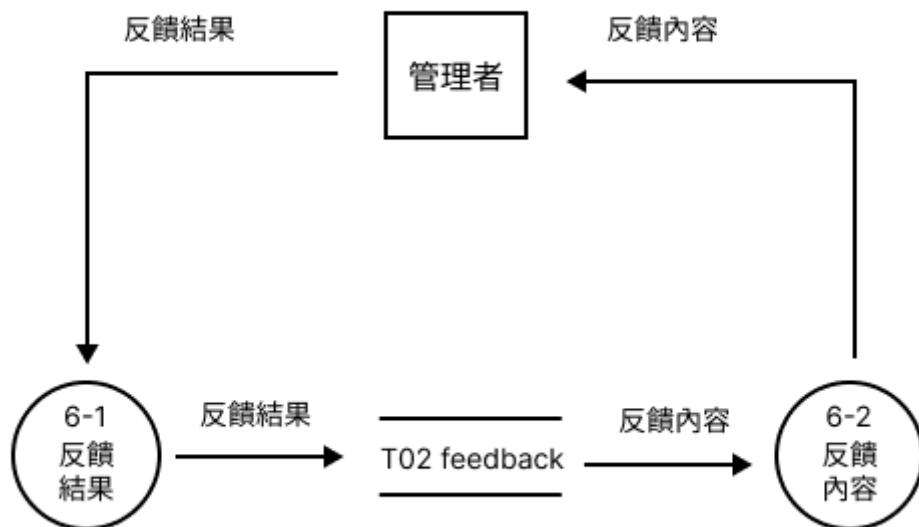
▲ 圖 6-1-5 DFD 圖 3



▲ 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-新增回饋

編號	4-1	功能名稱	新增回饋
功能說明	新增系統回饋資料		
輸入值	回饋內容		
輸出值	回饋成功訊息		

▼表 6-2-9 程序規格書-回饋結果

編號	4-2	功能名稱	回饋結果
功能說明	查看回饋結果		
輸入值	回饋結果		
輸出值	回饋結果內容		

▼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-回饋結果

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

▼表 6-2-15 程序規格書-回饋內容

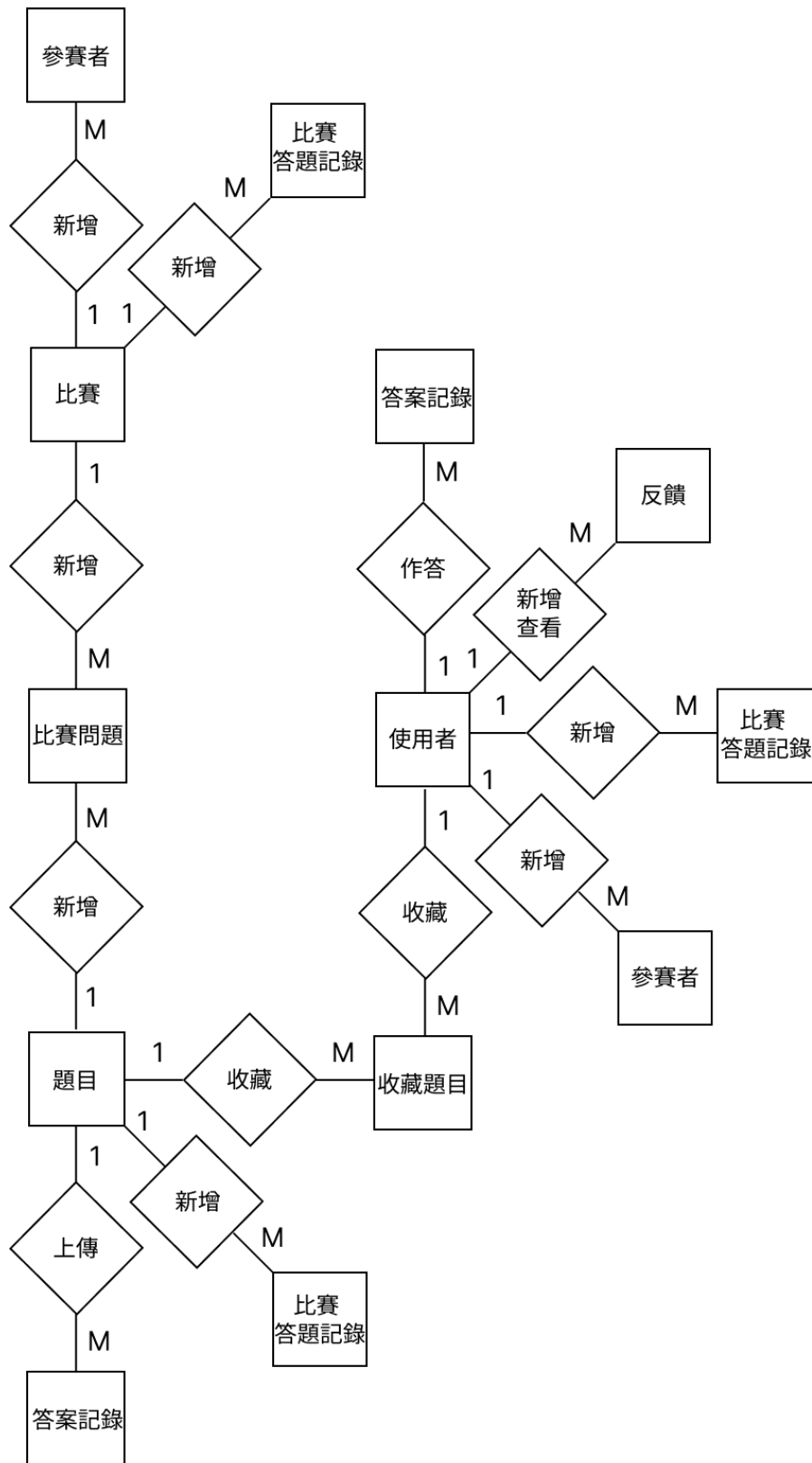
編號	6-2	功能名稱	回饋內容
功能說明	系統將回饋結果的內容發送給使用者		
輸入值	回饋內容		
輸出值	回饋內容資料		

▼表 6-2-16 程序規格書-使用者管理

編號	7	功能名稱	使用者管理
功能說明	管理者可修改使用者的權限		
輸入值	欲修改的使用者資料		
輸出值	新增成功訊息		

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

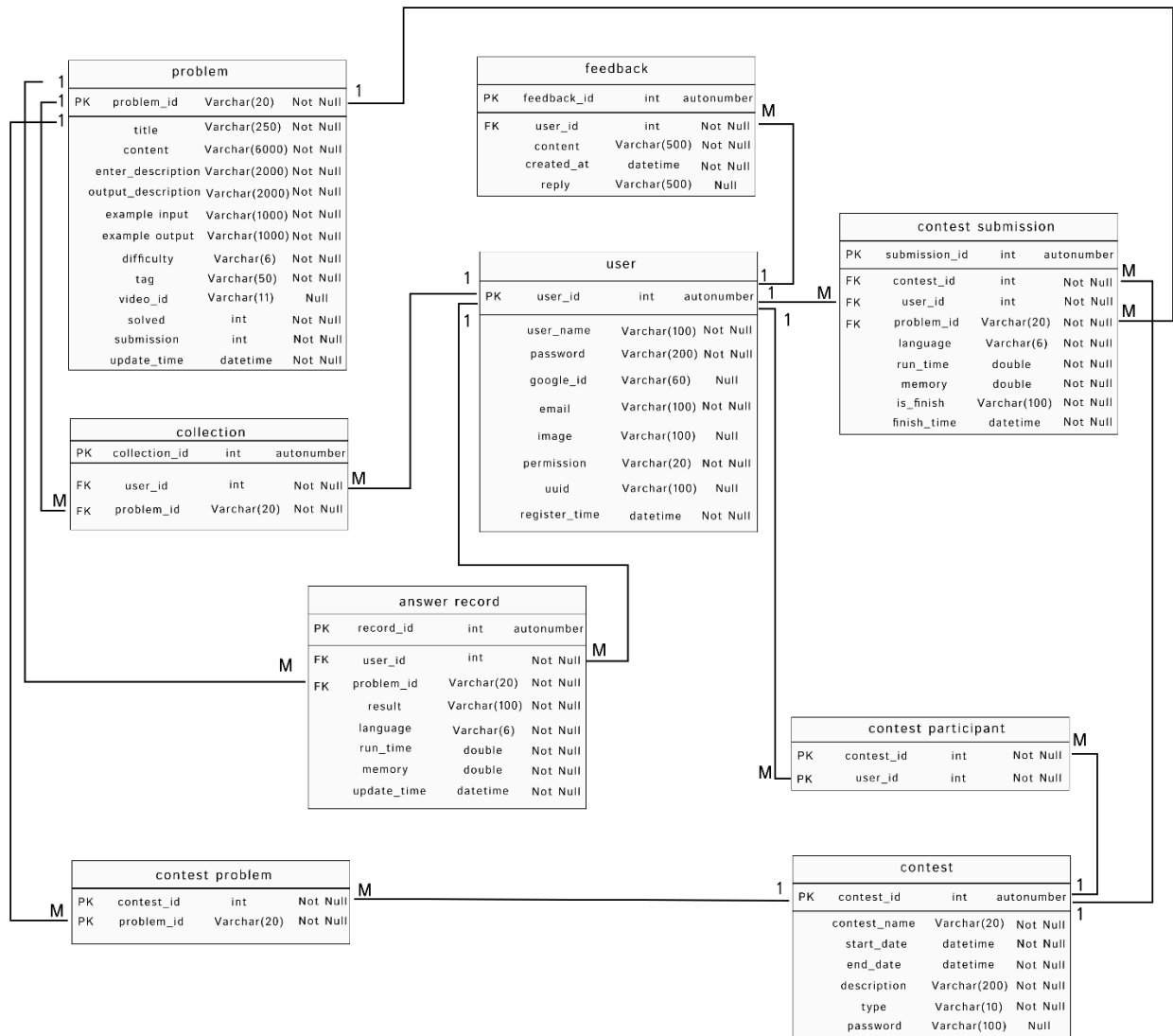
7-2 資料字典(Data dictionary)

▼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	回饋
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		✓	✓		
user_name	varchar	100	✓			
password	varchar	200	✓			
google_id	varchar	60				
email	varchar	100	✓			
image	varchar	100				
permission	varchar	20	✓			
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		✓	✓		
user_id	int		✓		✓	
content	varchar	500	✓			
created_at	datetime		✓			
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	✓	✓		
title	varchar	250	✓			
content	varchar	6000	✓			
enter description	varchar	2000	✓			
output description	varchar	2000	✓			
example input	varchar	1000	✓			
example output	varchar	1000	✓			
difficulty	varchar	6	✓			
tag	varchar	50	✓			
video_id	varchar	11				
solved	int		✓			
submission	int		✓			
update_time	datetime		✓			

▼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		✓	✓		
user_id	int		✓		✓	
problem_id	varchar	20	✓		✓	
result	varchar	100	✓			
language	varchar	6	✓			
run_time	double		✓			
memory	double		✓			
update_time	datetime		✓			

▼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		✓	✓		
user_id	int		✓		✓	
problem_id	varchar	20	✓		✓	

▼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		✓	✓		
problem_id	varchar	20	✓	✓		

▼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		✓	✓		
contest_name	varchar	20	✓			
start_date	datetime		✓			
end_date	datetime		✓			
description	varchar	200	✓			
type	varchar	10	✓			
password	varchar	100				

▼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		✓	✓		
user_id	int		✓	✓		

▼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		✓	✓		
contest_id	int		✓		✓	
user_id	int		✓		✓	
problem_id	varchar	20	✓		✓	
language	varchar	6	✓			
run_time	double		✓			
memory	double		✓			
is_finish	varchar	100	✓			
finish_time	datetime		✓			

第九章 程式

9-1 軟體架構與程式清單

▼表 9-1-1 前端程式清單

編號	檔案名稱	功能名稱	頁碼
1	templates\problem_list.html	首頁	33
2	templates\problem.html	題目作答介面	38
3	templates\rank.html	排行榜介面	44
4	templates\login.html	查詢電子郵件是否已註冊	45
5	templates\login_password.html	登入密碼介面	46
6	templates\forget_password.html	忘記密碼介面	47
7	templates\verify_forget_password.html	驗證忘記密碼郵件介面	48
8	templates\change_password.html	重設密碼介面	48
9	templates\feedback_create_form.html	用戶回饋表單	50
10	templates\feedback_history.html	回饋紀錄介面	50
11	templates\admin_dashboard.html	管理回饋表單	51
12	templates\reply_feedback_form.html	回復回饋表單	51
13	templates\user_data.html	用戶個人資料介面	52
14	templates\add_problem.html	新增題目介面	56
15	templates\manager_problem.html	管理題目介面	59
16	templates\manager_user.html	管理使用者介面	63
17	templates\answer_record.html	作答情況介面	65
18	templates\create_contest.html	創建比賽介面	66
19	templates\contest_joined.html	比賽介面	74
20	static\js\codemirro.js	控制程式碼輸入框上傳	80

▼表 9-1-2 後端程式清單

編號	檔案名稱	功能名稱	頁碼
1	app.py	系統主要功能	85
2	services\auth\app.py	使用者認證相關功能	91
3	services\feedback\app.py	回饋相關功能	99
4	services\user\app.py	用戶個人資料相關功能	103
5	services\manager\app.py	管理者相關功能	105
6	services\contest\app.py	比賽相關功能	109
7	utils\dolos.py	程式碼分析功能	117
8	crawler\add_problem.py	管理員上傳新的題目	118
9	crawler\submit.py	上傳題目到 Judge 平台	122

9-2 程式規格描述

▼表 9-2-1 程式清單

路徑	templates\problem_list.html
用途	首頁
部分程式碼	
<pre> <div class="container"> <div class="row mt-4"> <div class="col-2"> <select class="form-select" id="state" onchange="location.href=this.options[this.selectedIndex].value"> <option selected disabled>狀態</option> <option value="?page={{ page }}&state=*&onlinejudge={{ onlinejudge }}&difficulty={{ difficulty }}&tag={{ tag }}&search={{ search }}"> All </option> <option value="?page={{ page }}&state=accept&onlinejudge={{ onlineju dge }}&difficulty={{ difficulty }}&tag={{ tag }}&search={{ search }}" {% if (state=='accept') %}selected{% endif %}> 通過 </option> <option value="?page={{ page }}&state=error&onlinejudge={{ onlinejud ge }}&difficulty={{ difficulty }}&tag={{ tag }}&search={{ search }}" {% if (state=='error') %}selected{% endif %}> 錯誤 </option> <option value="?page={{ page }}&state=none&onlinejudge={{ onlinejud ge }}&difficulty={{ difficulty }}&tag={{ tag }}&search={{ search }}" {% if (state=='none') %}selected{% endif %}> 未做 </option> </select> </div> <div class="col-2"> <select class="form-select" id="onlinejudge" onchange="location.href=this.options[this.selectedIndex].value"> <option selected disabled>OnlineJudge</option> <option value="?page=1&state={{ state }}&onlinejudge=*&difficulty={{ difficulty }}&tag={{ tag }}&search={{ search }}"> </pre>	

```

        All
    </option>
    <option
        value="?page=1&state={{ state }}&onlinejudge=ZJ&difficulty={{
{ difficulty }}&tag={{ tag }}&search={{ search }}"
        {% if ( onlinejudge=='ZJ' ) %}selected{% endif %}>
        ZeroJudge
    </option>
    <option
        value="?page=1&state={{ state }}&onlinejudge=TIOJ&difficulty
={{ { difficulty }}&tag={{ tag }}&search={{ search }}"
        {% if ( onlinejudge=='TIOJ' ) %}selected{% endif %}>
        TIOJ
    </option>
    <option
        value="?page=1&state={{ state }}&onlinejudge=LeetCode&diffi
culty={{ { difficulty }}&tag={{ tag }}&search={{ search }}"
        {% if ( onlinejudge=='LeetCode' ) %}selected{% endif %}>
        LeetCode
    </option>
</select>
</div>
<div class="col-2">
    <select class="form-select" id="difficulty"
        onchange="location.href=this.options[this.selectedIndex].value">
        <option selected disabled>難度</option>
        <option
            value="?page={{ page }}&state={{ state }}&onlinejudge={{ onli
nejudge }}&difficulty=*&tag={{ tag }}&search={{ search }}">
            All
        </option>
        <option
            value="?page={{ page }}&state={{ state }}&onlinejudge={{ onli
nejudge }}&difficulty=Hard&tag={{ tag }}&search={{ search }}"
            {% if ( difficulty=='Hard' ) %}selected{% endif %}>
            Hard
        </option>
        <option
            value="?page={{ page }}&state={{ state }}&onlinejudge={{ onli
nejudge }}&difficulty=Normal&tag={{ tag }}&search={{ search }}"
            {% if ( difficulty=='Normal' ) %}selected{% endif %}>
            Middle
        </option>
        <option
            value="?page={{ page }}&state={{ state }}&onlinejudge={{ onli
nejudge }}&difficulty=Easy&tag={{ tag }}&search={{ search }}"

```

```

        {% if ( difficulty=='Easy' ) %}selected{% endif %}>
        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="tag" placeholder="標籤"
aria-label="Tag" id="tag"
        value="{{ tag }}">
        <input type="hidden" name="search" value="{{ search }}">
    </form>
</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 type="hidden" name="tag" value="{{ tag }}">
        <input class="form-control me-2" name="search" placeholder="名稱"
" 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]}}&source=main'">
<td class="text-success">
{% if item[14]=='Accepted' %}
<span style="color: #198754;"><strong class="text-nowrap">
✓ 已完成</strong></span>
{% elif (item[14]=='Rejected') or (item[14]=='Not Accept') or
(item[14]=='Wrong Answer')%}
<span style="color: #ffc107;"><strong class="text-
nowrap"> ! 未通過</strong></span>
{% endif %}
</td>
<td>
{% if item[0].split('-')[0] == 'ZJ' %}
Zero<br>Judge
{% elif item[0].split('-')[0] == 'TIOJ' %}
TIOJ
{% elif item[0].split('-')[0] == 'CAOJ' %}
CodeAlchemy
{% 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[13] }}<!--通過率-->
</td>
<td>{{ item[12].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 14.6a.5.5 0 0 1 .708 0l3 3a.5.5 0 0 1
0 .708l-10 10a.5.5 0 0 1 -.168 1.11l-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 1.06-1.528 3.821 3.821-
1.528 1.06-.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 1-1
0V6a.5.5 0 0 1 .5-.5m2.5 0a.5.5 0 0 1 .5.5v6a.5.5 0 0 1 1-1 0V6a.5.5 0 0 1 .5-.5m3 .5a.5.5
0 0 0 1 0v6a.5.5 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 4.059V13a1 1 0 0 0 1 1h6a1 1 0 0 0 1 1V4.059L11.882
4zM2.5 3h1V2h-1z" />
</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}}&tag={{tag}}&search={{search}}">第一頁</a>
</li>
{% else %}

```

```

</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={{onlineju
dge}}&difficulty={{difficulty}}&tag={{ tag }}&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}}&tag={{ tag }}&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
部分程式碼	
<pre> <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] }}
{{ data[0][1] }}</h1> </div> </div> </div> </div> </div> </pre>	

```

<!-- 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>
{% if source=='main' %}
<!-- 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>
{% endif %}
<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>
    {%endifor%}
</div>
{% if source=='main' %}
<!-- 教學影片 -->
<div class="tab-pane fade" id="menu2">
    <div class="col-12">
        {% if video_id == None %}
        <h3 class="justify-content-center">本題暫時沒有解題影
片</h3>
        {% else %}
        <iframe width="560" height="315"
src="https://www.youtube.com/embed/{{video_id}}"
        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>
        {% endif %}
    </div>
</div>
{% endif %}
</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-.708l-3.5-3.5a.5.5 0 0 0-.708 0" />
        </svg>測試</button>
<!-- Modal -->
<!-- 進度條視窗 -->
<!-- 正常上傳 -->
<div class="modal fade" id="loadingModal" data-bs-
backdrop="static" data-bs-keyboard="false"
    tabindex="-1" aria-labelledby="loadingModalLabel" aria-
hidden="true">
    <div class="modal-dialog">
        <div class="modal-content">
            <div class="modal-header">
                <h5 class="modal-title" id="loadingModalLabel">
測試中</h5>
            </div>
            <div class="modal-body">

```

```

        <div class="d-flex align-items-center">
            <strong>Loading...</strong>
            <div class="spinner-border ms-auto"
role="status" aria-hidden="true"></div>
        </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>
<!-- 沒登入 -->
<div class="modal fade" id="loginModal" data-bs-
backdrop="static" data-bs-keyboard="false"
        tabindex="-1" aria-labelledby="loginModalLabel" aria-
hidden="true">
    <div class="modal-dialog">
        <div class="modal-content">
            <div class="modal-header">
                <h5 class="modal-title" id="loginModalLabel">請
先登入</h5>
            </div>
            <div class="modal-body">
                您尚未登入，請先登入才能繼續操作。
            </div>
            <div class="modal-footer">
                <button type="button" class="btn btn-secondary"
                    data-bs-dismiss="modal">關閉</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>
{% if source=='main' %}
<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>
{% endif %}
<!-- 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	排行榜介面
部分程式碼	
<pre> <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 %} {% else %} {% endif %} <td>{{ 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> </pre>	

路徑	templates\login.html
用途	查詢電子郵件是否已註冊
部分程式碼	
<pre> <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"> <!-- 嵌入 Google 圖示 SVG --> 使用 Google 登入 </div> </div> </div> </div> </div> </div> </pre>	

路徑	templates\login_password.html
用途	登入密碼頁面
部分程式碼	
<pre> <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' %} 忘記密碼？ {% 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"); </pre>	

```

        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
部分程式碼	
<pre> <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> </pre>	

路徑	templates\verify_forget_password.html
用途	驗證忘記密碼郵件介面
部分程式碼	
<pre> <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> </pre>	

路徑	templates\change_password.html
用途	重設密碼介面
部分程式碼	
<pre> <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" </pre>	

```

        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	templates\feedback_create_form.html
用途	用戶回饋表單
部分程式碼	
<pre> <div class="container"> <div class="card border-0 shadow rounded-3 my-5"> <div class="card-body p-4 p-sm-5"> <h2>請提供您的回饋:</h2> <form action="/feedback/create" method="post"> <textarea class="form-control" name="feedback_content" rows="5" placeholder="在此輸入您的回饋"></textarea> <div class="btn-row"> <button type="submit" class="btn btn-primary">提交回饋 </button> 回饋紀錄 </div> </form> </div> </div> </div> </pre>	

路徑	templates\feedback_history.html
用途	回饋紀錄頁面
部分程式碼	
<pre> <div class="container"> {% if feedback_history %} <table> <tr> <th>回饋內容</th> <th>提交時間</th> </pre>	

```

        <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>沒有回饋紀錄</p>
{% endif %}
</div>

```

路徑	templates\admin_dashboard.html
用途	管理回饋表單
部分程式碼	
<pre> <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>使用者名稱: {{ feedback[1] }}</p> <p>回饋內容: {{ feedback[2] }}</p> <p>提交時間: {{ feedback[3] }}</p> <p>回覆內容: {{ 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> </pre>	

路徑	templates\reply_feedback_form.html
用途	回復回饋表單
部分程式碼	
<pre> <div class="reply-form"> <form action="/feedback/reply_feedback" method="post"> <p>使用者名稱: {{ feedback[1] }}</p> </pre>	

```

<p><strong>回饋內容:</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	templates\user_data.html
用途	用戶個人資料介面
部分程式碼	
<pre> <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 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 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> <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"> </pre>	

```

        </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 -->
                
                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	templates\add_problem.html
用途	新增題目介面
部分程式碼	
<pre> <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> </form> </div> </pre>	

```

<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">測試資料 輸入#1:</h3>
<div class="mb-3">
  <textarea class="form-control" name="test_input[]" id="" aria-
describedby="basic-addon2"
    placeholder="輸入#1" ></textarea>
</div>
<h3 class="mt-4">測試資料 輸出#1:</h3>
<div class="mb-3">
  <textarea class="form-control" name="test_output[]" id="" aria-
describedby="basic-addon2"
    placeholder="輸出#1" ></textarea>
</div>
<div id="additional-rows"></div>

<button class="btn btn-primary mt-4" id="add-row-btn" type="button">新增
一筆</button>

<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 4.39A1.5 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>
    let count = 1;

```

```

document.getElementById("add-row-btn").addEventListener("click", function ()
{
    count++;

    // 創建新的輸入/輸出行
    const newInputGroup = `
        <h3 class="mt-4">測試資料 輸入#${count}</h3>
        <div class="mb-3">
            <input type="text" class="form-control" name="test_input[]"
placeholder="輸入#${count}" />
        </div>
        <h3 class="mt-4">測試資料 輸出#${count}</h3>
        <div class="mb-3">
            <input type="text" class="form-control" name="test_output[]"
placeholder="輸出#${count}" />
        </div>
    `;

    // 將新行加入到頁面中
    document.getElementById("additional-
rows").insertAdjacentHTML('beforeend', newInputGroup);
});

document.getElementById('confirm_save').addEventListener('click', function () {
    document.querySelector('form').submit();
});

</script>

```

路徑	templates\manager_problem.html
用途	管理題目介面
部分程式碼	
<pre> <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:
</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" </pre>	

```

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">測試資料輸入:</h3>
<div class="mb-3">
  <input type="text" class="form-control"
name="test_input" id=""
      aria-describedby="basic-addon2" placeholder="" />
</div>
<h3 class="mt-4">測試資料輸出:</h3>
<div class="mb-3">
  <input type="text" class="form-control"
name="test_input" id=""
      aria-describedby="basic-addon2" placeholder="" />
</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 2h2v3h2z" />
                <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>

```

路徑	templates\manager_user.html
用途	管理使用者介面
部分程式碼	
<pre> <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> </pre>	


```

        </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>

```

路徑	templates\answer_record.html
用途	作答情況介面
部分程式碼	
<pre> <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>作答編號</td> <td>用戶編號</td> <td>用戶姓名</td> <td>作答結果</td> <td>程式語言</td> <td>執行時間</td> <td>記憶體</td> <td>上傳日期</td> </tr> </thead> <tbody> {% for item in data %} <tr> <td>{{ item[0] }}</td> <td>{{ item[1] }}</td> <td>{% if item[2] is none %} {% else %} {% endif %} <td>{{ item[3] }}</td> <td>{{ item[4] }}</td> <td>{{ item[5] }}</td> <td>{{ item[6] }}MS</td> <td>{{ item[7] }}MB</td> <td>{{ item[8] }}</td> </tr> {% endfor %} </tbody> </table> </pre>	

</div>

</div>

路徑	templates\create_contest.html
用途	創建比賽介面
部分程式碼	
<pre><div class="container-fluid"> <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"> <form id="competitionForm" action="/contest/create" method="post"> <h1>創建比賽</h1> <hr class="my-4"> <div class="form"> <div class="form-group row mb-3"> <label for="contest_name" class="col-sm-3 col-form- label">比賽名稱:</label> <div class="col-sm-9"> <input type="text" class="form-control" id="contest_name" placeholder="請輸入比賽名稱" name="contest_name" required> </div> </div> <div class="form-group row mb-3"> <label for="description" class="col-sm-3 col-form- label">比賽說明:</label> <div class="col-sm-9"> <input type="text" class="form-control" id="description" placeholder="請輸入比賽說明" name="description" style="width: 100%;" required> </div> </div> <div class="form-group row mb-3"> <label for="state" class="col-sm-3 col-form-label">比 賽類型:</label> <div class="col-sm-9"> <select class="form-select" id="state" name="state"></pre>	

```

        <option value="public">公開</option>
        <option value="private">私人</option>
    </select>
</div>
</div>

<!--當比賽類型為 private 時出現-->
<div class="form-group row mb-3" id="passwordField"
style="display: none;">
    <label for="contest_password" class="col-sm-3 col-
form-label">比賽密碼:</label>
    <div class="col-sm-9">
        <input type="password" class="form-control"
id="contest_password" placeholder="請輸入比賽密碼" name="contest_password">
    </div>
</div>

    <div class="form-group row mb-3"
id="confirmPasswordField" style="display: none;">
        <label for="confirm_password" class="col-sm-3 col-
form-label">確認密碼:</label>
        <div class="col-sm-9">
            <input type="password" class="form-control"
id="confirm_password" placeholder="請輸入確認密碼" name="confirm_password">
        </div>
    </div>
    <!--當比賽類型為 private 時出現-->

    <div class="form-group row mb-3">
        <label for="startTime" class="col-sm-3 col-form-
label">開始時間:</label>
        <div class="col-sm-9">
            <input type="datetime-local" class="form-control"
id="startTime" name="startTime" required>
        </div>
    </div>

    <div class="form-group row mb-3">
        <label for="endTime" class="col-sm-3 col-form-
label">結束時間:</label>
        <div class="col-sm-9">
            <input type="datetime-local" class="form-control"
id="endTime" name="endTime" required>
        </div>
    </div>

```

```

<hr class="my-4">

<!-- 新增題目表格 -->
<div class="mb-3">
  <label class="form-label">題目列表:</label>
  <div>
    <table class="table">
      <thead>
        <tr>
          <th>#</th>

          <th>Name</th>
          <th>Difficulty</th>
          <th>Remove</th>
        </tr>
      </thead>
      <tbody id="questionTableBody">
        <!-- 這裡會動態添加所選擇的題目 -->
      </tbody>
    </table>
    <input href="#" class="btn btn-success w-100"
type="button" value="+ 新增題目 "
                                data-bs-toggle="modal" data-bs-
target="#loginModal"></span>
  </div>
</div>

<!-- 跳出視窗內容 -->
<div class="modal fade" id="loginModal">
  <div class="modal-dialog modal-xl">
    <div class="modal-content">
      <!-- Header -->
      <div class="modal-header">
        <h3>題目列表</h3>
        <button type="button" class="btn-close"
          data-bs-dismiss="modal"></button>
      </div>

      <!-- Body -->
      <div class="modal-body">
        <form>
          <table class="table">
            <thead>
              <tr>

```

```

</th>
</th>
</th>
</th>
scope="col">Action</th>
</tr>
</thead>
<tbody id="contestTableBody">
  <!-- 表格內容將在這裡使用
  </tbody>
</table>
<!-- 分頁導航 -->
<nav aria-label="Page navigation
example" class="d-flex justify-content-center">
  <ul class="pagination"
id="pagination">
    <!-- 分頁按鈕將動態插入到這裡
-->
  </ul>
</nav>
</form>
</div>
</div>
</div>
</div>
</div>
<hr class="my-4">
<div class="mb-3 text-center">
  <span class="value">
    <!-- 取消按鈕 -->
    <button class="btn btn-outline-secondary text-
uppercase fw-bold me-2" type="button" onclick="window.location.href='/'">取消
  </button>
    <!-- 創建比賽按鈕 -->
    <button class="btn btn-primary btn-login text-
uppercase fw-bold" type="submit" value="create">創建比賽</button>
  </span>

```

```

        </div>
    </form>
</div>
</div>
</div>
</div>
</div>

<script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
<script>
    // 取得當前時間並格式化為 datetime-local 格式
    const now = new Date();
    const yyyy = now.getFullYear();
    const mm = String(now.getMonth() + 1).padStart(2, '0'); // 月份從 0 開始
    const dd = String(now.getDate()).padStart(2, '0');
    const hh = String(now.getHours()).padStart(2, '0');
    const min = String(now.getMinutes()).padStart(2, '0');

    // 設定 datetime-local 的最小值
    const minDateTime = `${yyyy}-${mm}-${dd}T${hh}:${min}`;
    document.getElementById('startTime').setAttribute('min', minDateTime);
    document.getElementById('endTime').setAttribute('min', minDateTime);

    // 按下創建比賽後
    $(document).ready(function () {
        const selectedProblems = []; // 儲存已選擇的題目 ID

        function simplifyContent(content, maxLength) {
            if (content.length > maxLength) {
                return content.substring(0, maxLength) +
                    '<a href="#" class="more-link" data-content="' +
encodeURIComponent(content) + '">...more</a>';
            }
            return content;
        }

        // 比賽類型選擇框的變化
        $('#state').change(function() {
            var selectedValue = $(this).val();
            if(selectedValue === "private") {
                $('#passwordField').show();
                $('#confirmPasswordField').show();
            } else {
                $('#passwordField').hide();
                $('#confirmPasswordField').hide();
            }
        });
    });

```

```

    }
  });

  //提交表單前進行密碼一致性檢查
  $('#competitionForm').submit(function (e) {
    var password = $('#contest_password').val();
    var confirmPassword = $('#confirm_password').val();
    if($('#state').val() === "private" && password !== confirmPassword) {
      alert("密碼和確認密碼不匹配");
      e.preventDefault(); // 阻止表單提交
    }
  });
  /***/

  // 載入並顯示問題清單和分頁控件
  function loadTable(page) {
    $.ajax({
      url: '/contest/get_problems',
      type: 'GET',
      data: {
        page: page,
        per_page: 10
      },
      success: function (response) {
        var contestTableBody = $("#contestTableBody");
        contestTableBody.empty();

        response.data.forEach(function (problem, index) {
          var problem_id = problem[0];
          var title = problem[1];
          var content = simplifyContent(problem[2], 100);
          var difficulty = problem[3];
          var isAdded = selectedProblems.includes(problem_id); // 檢
查當前題目是否已被添加
          var addButtonClass = isAdded ? 'btn-success disabled' : 'btn-
primary';
          var addButtonText = isAdded ? '<i class="fas fa-
check"></i>' : 'Add'; // 如果已新增，顯示勾選符號
          var row = `<tr>
            <td>${problem_id}</td>
            <td>${title}</td>
            <td>${content}</td>
            <td>${difficulty}</td>
            <td><button class="btn ${addButtonClass} add-btn"
type="button" data-title="${title}" data-problem-id="${problem_id}" data-

```



```

difficulty="${difficulty}">${addButtonText}</button></td>
</tr>`;

    contestTableBody.append(row);
});

var pagination = $('#pagination');
pagination.empty();

var total_pages = response.total_pages;
var page = response.page;

var startPage = Math.max(page - 2, 1);
var endPage = Math.min(page + 2, total_pages);

pagination.append('<li class="page-item ' + (page === 1 ?
'disabled' : ") + '" data-page="1"><a class="page-link" href="#"><<</a></li>');

    for (let i = startPage; i <= endPage; i++) {
        var activeClass = page === i ? 'active' : "";
        var pageItem = '<li class="page-item ' + activeClass + '" data-
page="' + i + '"><a class="page-link" href="#">' + i + '</a></li>';
        pagination.append(pageItem);
    }

    pagination.append('<li class="page-item ' + (page ===
total_pages ? 'disabled' : ") + '" data-page="1" + total_pages + '"><a class="page-link"
href="#">>></a></li>');
    }
});
}

loadTable(1);

$('#pagination').on('click', 'li', function (e) {
    e.preventDefault();
    var page = $(this).data('page');
    loadTable(page);
});

//$("#contestTableBody").on('click', '.add-btn:not(.disabled)', function () {
$('#contestTableBody').on('click', '.add-btn', function () {
    var problemTitle = $(this).data('title');
    var problemId = $(this).data('problem-id');
    var difficulty = $(this).data('difficulty');
    var addButton = $(this);

```

```

        if (!selectedProblems.includes(problemId)) { // 檢查以避免重複添加
            selectedProblems.push(problemId); // 將目前題目 ID 新增至已選擇
            題目數組

            var row = `<tr class="added-question" data-problem-
id="${problemId}">
                <td></td>
                <td>${problemTitle}</td>
                <td>${difficulty}</td>
                <td class="text-center"><button class="btn btn-danger remove-
btn" data-problem-id="${problemId}"><i class="fas fa-trash-alt"></i></button></td>
            </tr>`;
            $("#questionTableBody").append(row);

            // Change the button to a check mark and disable it
            $(this).removeClass('btn-primary').addClass('btn-success
disabled').html('<i class="fas fa-check"></i>');
        }

        //$('#loginModal').modal('hide');
    });

    $('#pagination').on('click', 'li', function (e) {
        e.preventDefault();
        var page = $(this).data('page');
        loadTable(page);
    });

    $('#competitionForm').submit(function (e) {
        var selectedProblemIds = $('<div>added-question</div>').map(function () {
            return $(this).data('problem-id');
        }).get();

        if (selectedProblemIds.length === 0) {
            alert("請選擇至少一個題目");
            e.preventDefault(); // 停止提交表單
            return;
        }

        $('<input>').attr({
            type: 'hidden',
            name: 'problem_ids',
            value: selectedProblemIds.join(',')
        }).appendTo('#competitionForm');
    });

```

```

});

// 刪除表格中的問題
$('#questionTableBody').on('click', '.remove-btn', function () {
    var row = $(this).closest('tr');
    var index = row.index();
    var problem_id = selectedProblems[index];

    // 刪除所選題目並移除該行
    selectedProblems.splice(index, 1);
    row.remove();

    // 更新行號
    $('#questionTableBody tr').each(function (i) {
        $(this).find('td:first').text(i + 1);
    });

    // 恢復互動視窗中的按鈕狀態
    $('#contestTableBody button[data-problem-id="' + problem_id +
    "']").removeClass('btn-success disabled').addClass('btn-primary').text('Add');
});
});
</script>

```

路徑	templates\contest_join.html
用途	比賽介面
部分程式碼	
<pre> <div class="container"> <div class="row mt-4"> <div class="col"> <select class="form-select" id="typeSelect" onchange="updateFilters()"> <option value="" {% if contest_type is none or contest_type == '*' %}selected disabled{% endif %}> 設定 </option> <option value="" {% if contest_type == '*' %}selected{% endif %}> 全部 </option> <option value="public" {% if contest_type == 'public' %}selected{% endif %}> 公開 </option> <option value="private" {% if contest_type == 'private' </pre>	

```

%}selected{% endif %}>
        私人
    </option>
</select>
</div>
<!--語言下拉式選單註解
<div class="col-3">
    <select class="form-select" id="onlinejudge"
        onchange="location.href=this.options[this.selectedIndex].value">
        <option selected>語言</option>
        <option
            value="?page={{ page }}&state={{ state }}&onlinejudge=*&diffi
culty={{ difficulty }}&search={{ search }}">
            All</option>
        <option
            value="?page={{ page }}&state={{ state }}&onlinejudge=ZeroJud
ge&difficulty={{ difficulty }}&search={{ search }}">
            python</option>
        <option
            value="?page={{ page }}&state={{ state }}&onlinejudge=LeetCo
de&difficulty={{ difficulty }}&search={{ search }}">
            C</option>
        <option
            value="?page={{ page }}&state={{ state }}&onlinejudge=LeetCo
de&difficulty={{ difficulty }}&search={{ search }}">
            C++</option>
        <option
            value="?page={{ page }}&state={{ state }}&onlinejudge=LeetCo
de&difficulty={{ difficulty }}&search={{ search }}">
            python</option>
    </select>
</div>
-->
<div class="col">
    <select class="form-select" id="stateSelect" onchange="updateFilters()">
        <option value="*" {% if state is none or state == '*' %}>selected
disabled{% endif %}>
            狀態
        </option>
        <option value="*" {% if state == '*' %}>selected{% endif %}>
            全部
        </option>
        <option value="not_started" {% if state == 'not_started'
%}>selected{% endif %}>
            尚未開始

```

```

        </option>
        <option value="ongoing" {% if state == 'ongoing' %}selected{%
endif %}>
            進行中
        </option>
        <option value="finished" {% if state == 'finished' %}selected{%
endif %}>
            已結束
        </option>
    </select>
</div>

<div class="col">
    <form class="d-flex" method="get">
        <input type="hidden" name="page" value="{{page}}">
        <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>

{% if no_contest_message %}
    <div class="text-center mt-3">
        <strong>{{ no_contest_message }}</strong>
    </div>
{% endif %}

</div>
<!-- ----- -->
<!-- 題目列表 -->
<!-- ----- -->
<div>
    {% for contest in contests %}
    <div class="competition-box">
        <div class="competition-info">
            <div class="title">
                {% if contest[5] == 'private' %}
                <!-- 顯示黃色小鎖符號 -->
                <span style="color: yellow;">🔒 </span>
                {% endif %}
                {{ contest[1] }}

```

```

</div>
<div class="description">{{ contest[4] }}</div>

<!-- 顯示錯誤或成功訊息 -->
{% with messages = get_flashed_messages() %}
    {% if messages %}
        <div class="alert alert-warning">
            {{ messages[0] }} <!-- 可以擴展以顯示多個訊息 -->
        </div>
    {% endif %}
{% endwith %}

<!-- 比賽列表的按鈕區域 -->
<form id="contestForm-{{ contest[0] }}"
action="{{ url_for('contest_bp.join_contest') }}" method="POST">
    <input type="hidden" name="contest_id"
value="{{ contest[0] }}">
    <input type="hidden" name="contest_type"
value="{{ contest[5] }}"> <!-- 用於傳遞比賽類型 -->
    <input type="hidden" id="passwordField-{{ contest[0] }}"
name="contest_password" value=""> <!-- 密碼欄位 -->

    {% if contest[2] > current_time %}
        <button type="button" class="join-button btn"
style="background-color: #8d8d8d; color: #000000;" disabled>尚未開始</button>
    {% elif current_time > contest[3] %}
        <button type="button" class="join-button btn" data-
type="{{ contest[5] }}" style="background-color: #b60202; color: #FFFFFF;"
onclick="handleContestClick('{{ contest[0] }}', '{{ contest[5] }}')">觀看比賽結果
</button>
    {% elif contest[6] == "joined" %}
        <button type="button" class="join-button btn btn-purple" data-
type="{{ contest[5] }}" onclick="handleContestClick('{{ contest[0] }}',
'{{ contest[5] }}')">繼續完成比賽</button>
    {% else %}
        <button type="button" class="join-button btn" data-
type="{{ contest[5] }}" onclick="handleContestClick('{{ contest[0] }}',
'{{ contest[5] }}')">加入比賽</button>
    {% endif %}
</form>
</div>
<div class="competition-details">
    <p style="margin-bottom: 10px;">參加人數：{{ contest[7] }}</p>
    <p style="margin-bottom: 10px;">開始時間：{{ contest[2] }}</p>

```

```

        <p style="margin-bottom: 10px;">結束時間： {{ contest[3] }}</p>
        <p style="color: red; margin-bottom:
0;"><strong>{{ contest[8] }}</strong></p>
    </div>
</div>

<!-- 密碼輸入的 Modal -->
<div class="modal fade" id="passwordModal" tabindex="-1" role="dialog"
aria-labelledby="passwordModalLabel" aria-hidden="true">
    <div class="modal-dialog" role="document">
        <div class="modal-content">
            <div class="modal-header">
                <h5 class="modal-title" id="passwordModalLabel">輸入比
賽密碼</h5>
                <button type="button" class="btn-close" data-bs-
dismiss="modal" aria-label="Close"></button>
            </div>
            <div class="modal-body">
                <input type="password" class="form-control"
id="contestPassword" placeholder="請輸入比賽密碼">
                <div id="passwordError" class="text-danger" style="display:
none;">密碼輸入錯誤</div>
            </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"
onclick="submitPassword(window.currentContestId)">確認</button>
            </div>
        </div>
    </div>
</div>
</div>
{% endfor %}
</div>

<!-- 分頁按鈕 -->
<div class="d-flex justify-content-center mt-5">
    <nav aria-label="Page navigation">
        <ul class="pagination">
            <!-- 第一頁按鈕 -->
            <li class="page-item">
                <a class="page-link"
href="?page=1&state={{ state }}&onlinejudge={{ onlinejudge }}&contest_type={{ co
ntest_type }}&search={{ search }}" aria-label="First">

```

```

        <span aria-hidden="true">&laquo;</span>
    </a>
</li>
    {% for p in range(1, total_pages + 1) %}
    <li class="page-item {% if p == page %}active{% endif %}">
        <a class="page-link"
href="?page={{ p }}&state={{ state }}&onlinejudge={{ onlinejudge }}&contest_type=
{{ contest_type }}&search={{ search }}">{{ p }}</a>
    </li>
    {% endfor %}
    <!-- 最後一頁按鈕 -->
    <li class="page-item">
        <a class="page-link"
href="?page={{ total_pages }}&state={{ state }}&onlinejudge={{ onlinejudge }}&cont
est_type={{ contest_type }}&search={{ search }}" aria-label="Last">
            <span aria-hidden="true">&raquo;</span>
        </a>
    </li>
</ul>
</nav>
</div>

<!-- 沒登入 -->
<div class="modal fade" id="loginModal" data-bs-backdrop="static" data-bs-
keyboard="false"
tabindex="-1" aria-labelledby="loginModalLabel" aria-hidden="true">
    <div class="modal-dialog">
        <div class="modal-content">
            <div class="modal-header">
                <h5 class="modal-title" id="loginModalLabel">請先登入</h5>
            </div>
            <div class="modal-body">
                您尚未登入，請先登入才能繼續操作。
            </div>
            <div class="modal-footer">
                <button type="button" class="btn btn-secondary"
                    data-bs-dismiss="modal">關閉</button>
            </div>
        </div>
    </div>
</div>
</div>
</div>

```


路徑	static\js\codemirro.js
用途	控制程式碼輸入框上傳
部分程式碼	
<pre>// 載入 CodeMirror 編輯器 var editor = CodeMirror.fromTextArea(document.getElementById('editor'), { // 設定行號 lineNumbers: true, // 設定縮排單位 indentUnit: 4, // 初始模式（默認為 C） mode: 'python' }); // 顯示和隱藏 loading 視窗的函數 function showLoading() { document.getElementById("staticBackdrop").style.display = "block"; } function hideLoading() { document.getElementById("staticBackdrop").style.display = "none"; // 手動移除 backdrop var backdrops = document.getElementsByClassName('modal-backdrop'); while (backdrops.length > 0) { backdrops[0].parentNode.removeChild(backdrops[0]); } } // 取得下拉選單 var selectLanguageButton = document.getElementById('select_language_button'); var mode = 'python'; // 監聽點擊事件 selectLanguageButton.addEventListener('click', function (event) { // 確保點擊的是 a 標籤 if (event.target.tagName === 'A') { // 取得選項中的 data-mode 屬性值 mode = event.target.getAttribute('data-mode'); // 設定編輯器模式 editor.setOption('mode', mode); } }); // 定義變數以保存使用者 ID var userId = null;</pre>	

```

document.addEventListener("DOMContentLoaded", function () {

    // 發送請求以獲取使用者 ID
    var xhr = new XMLHttpRequest();
    xhr.open("GET", "/get_user_id", true); // 這裡的路徑是之前後端定義的接口
    xhr.send();

    xhr.onreadystatechange = function () {
        if (xhr.readyState === XMLHttpRequest.DONE) {
            if (xhr.status === 200) {
                var response = JSON.parse(xhr.responseText);
                userId = response.user_id;
                // console.log("使用者 ID: ", userId);
                // 你可以在這裡使用 userId 或將其保存到全局變數中以供後續使用
            }
        }
    };
});

// 監聽按鈕點擊事件
document.getElementById("test_btn").addEventListener("click", function () {
    // 隱藏之前的結果
    document.getElementById("passed").style.display = "none";
    document.getElementById("failed").style.display = "none";
    var loginModal = new bootstrap.Modal(document.getElementById('loginModal'));
    var loadingModal = new
bootstrap.Modal(document.getElementById('loadingModal'));

    if (!userId) {
        // 如果沒有取得 userId，顯示登入提示模態視窗
        loginModal.show();
        return; // 阻止請求發送
    } else {
        // 如果已登入，顯示測試中的模態視窗
        loadingModal.show();

        // 這裡可以繼續執行你的後續請求邏輯
        var type = 'test'; // 預設是 test
        var urlParams = new URLSearchParams(window.location.search);
        var source = urlParams.get('source');
        var contestId = urlParams.get('contest_id');
        var problem_id = document.getElementById('problem_id').innerHTML;
        var language = mode;

        // 獲取編輯器中的程式碼

```

```

var code = editor.getValue();

// 構建表單數據
var formData = new FormData();
formData.append("user_id", userId); // 添加使用者 ID
formData.append("type", type);
formData.append("problem_id", problem_id);
formData.append("language", language);
formData.append("code", code);
formData.append("source", source);
if (source === 'contest' && contestId) {
    formData.append("contest_id", contestId);
}
// 發送 POST 請求到伺服器
var xhr = new XMLHttpRequest();
xhr.open("POST", "/problem");
xhr.send(formData);

// 處理伺服器返回的數據
xhr.onreadystatechange = function () {
    if (xhr.readyState === XMLHttpRequest.DONE) {
        if (xhr.status === 200) {
            loadingModal.hide(); // 請求完成後隱藏 loading 視窗
            var responseData = JSON.parse(xhr.responseText);
            var result = responseData.result;
            var message = responseData.message;
            var runTime = responseData.run_time;
            var memory = responseData.memory;

            if (result) {
                document.getElementById("passed_status").innerText = "測試通
過";
                document.getElementById("passed_run_time").innerText = "執行
時間：" + runTime + "毫秒";
                document.getElementById("passed_memory").innerText = "記憶
體使用量：" + memory + "MB";
                document.getElementById("passed").style.display = "block";
            } else {
                document.getElementById("failed_status").innerText = "測試失
敗";
                document.getElementById("failed_error_reason").innerText = "錯
誤原因：" + message;
                document.getElementById("failed").style.display = "block";
            }
        }
    }
}

```

```

        } else {
            console.error('請求失敗');
        }
    }
};
});
//上傳
// 監聽按鈕點擊事件
document.getElementById("upload_btn").addEventListener("click", function () {
    // 隱藏之前的結果
    document.getElementById("passed").style.display = "none";
    document.getElementById("failed").style.display = "none";
    // 初始化模態視窗
    var loginModal = new bootstrap.Modal(document.getElementById('loginModal'));
    var loadingModal = new
bootstrap.Modal(document.getElementById('loadingModal'));

    // 檢查使用者是否已登入
    if (!userId) {
        loginModal.show(); // 如果沒有取得 userId，顯示登入提示模態視窗
        return; // 阻止請求發送
    }

    // 如果已登入，顯示測試中的模態視窗
    loadingModal.show();

    // 獲取必要的參數
    var type = 'upload';
    var urlParams = new URLSearchParams(window.location.search);
    var problem_id = document.getElementById('problem_id').innerText;
    var title = document.getElementsByClassName('display-6 mt-4')[0].innerText;
    var language = mode; // mode 應該是你代碼其他地方定義的變數
    var source = urlParams.get('source');
    var contestId = urlParams.get('contest_id');
    var code = editor.getValue(); // 從編輯器中獲取程式碼

    // 構建表單數據
    var formData = new FormData();
    formData.append("type", type);
    formData.append("problem_id", problem_id);
    formData.append("title", title);
    formData.append("language", language);
    formData.append("code", code);

```

```

formData.append("source", source);

// 如果是來自 contest 的請求，添加 contest_id
if (source === 'contest' && contestId) {
    formData.append("contest_id", contestId);
}

// 發送 POST 請求到伺服器
var xhr = new XMLHttpRequest();
xhr.open("POST", "/problem", true);

// 處理伺服器返回的數據
xhr.onreadystatechange = function () {
    if (xhr.readyState === XMLHttpRequest.DONE) {
        if (xhr.status === 200) {
            // 隱藏 loading 視窗
            loadingModal.hide();

            // 處理伺服器返回的響應
            var responseData = JSON.parse(xhr.responseText);
            var result = responseData.result;
            var message = responseData.message;
            var runTime = responseData.run_time;
            var memory = responseData.memory;

            if (result) {
                // 測試通過，顯示成功信息
                document.getElementById("passed_status").innerText = "測試通過";
                document.getElementById("passed_run_time").innerText = "執行時間：" + runTime + "毫秒";
                document.getElementById("passed_memory").innerText = "記憶體使用量：" + memory + "MB";
                document.getElementById("passed").style.display = "block"; // 顯示 passed 的信息
            } else {
                // 測試失敗，顯示錯誤信息
                document.getElementById("failed_status").innerText = "測試失敗";
                document.getElementById("failed_error_reason").innerText = "錯誤原因：" + message;
                document.getElementById("failed").style.display = "block"; // 顯示 failed 的信息
            }
        } else {

```

```

        // 請求失敗，隱藏 loading 視窗並顯示錯誤信息
        loadingModal.hide();
        console.error('請求失敗，狀態碼：', xhr.status);
        alert('提交失敗，請稍後再試。');
    }
}
};

// 發送表單數據
xhr.send(formData);
});

```

路徑	app.py
用途	系統主要功能
部分程式碼	
<pre> #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) tag = request.args.get('tag', '', 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+=f"""AS state ON p.problem_id = state.problem_id""" </pre>	

```

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 tag != "":
        condition += f'p.tag LIKE "%{tag}%" AND '
    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
    data=db.get_data(sql_problem_command)
    # 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,tag=tag,search=search)

# 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
        type=data.get('type')
        problem_id = data.get('problem_id')
        language = data.get('language')
        code = data.get('code')
        source = data.get('source')
        contest_id = data.get('contest_id')

        try:
            title = data.get('title')
        except:
            title = None

        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, str(session['User_id']))
elif "TIOJ" in file_name:
    score = TIOJ_submit(file_name, str(session['User_id']))
elif "CAOJ" in file_name:
    score = CodeAlchemy_submit(file_name, str(session['User_id']), title)

result = False
message = "測試失敗"
run_time = 0
memory = 0
ensue = "Rejected"
mime_to_language = {
    "text/x-c++src": "C++",
    "python": "Python",
    "text/x-csrc": "C",
    "text/x-java": "Java",
    # 可以根據需要繼續添加其他對應項
}
language = mime_to_language.get(language, "Unknown")
if score and score[2] == 'Accepted':
    result = True
    message = "測試成功"
    run_time = score[3].replace('ms', '')
    memory = score[4].replace('MB', '')
    ensue = score[2]
# 確認提交來源並插入到正確的資料表
print(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]}')
    ")
    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, source=source)
```

```
@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)
@app.errorhandler(404)
def page_not_found(error):
    return render_template('page_not_found.html')
@app.errorhandler(Exception)
def page_not_found(error):
    return render_template('page not found.html')

```

路徑	services\auth\app.py
用途	使用者認證相關功能
部分程式碼	
<pre> # 傳送驗證電子郵件 app.config['SECRET_KEY'] = 'itismysecretkey' app.config['MAIL_SERVER']='smtp.gmail.com' app.config['MAIL_PORT'] = 465 app.config['MAIL_USERNAME'] = MAIL_USERNAME app.config['MAIL_PASSWORD'] = MAIL_PASSWORD app.config['MAIL_USE_TLS'] = False app.config['MAIL_USE_SSL'] = True mail = Mail(app) </pre>	

```

# 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 庫的 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():
    try:
        # 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 帳戶頁面
            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')
        except:
            return redirect('/auth/login')

# 連結 Google 帳戶路由
@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 帳戶頁面
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	services\feedback\app.py
用途	回饋相關功能
部分程式碼	
<pre> # 產生回饋服務藍圖 feedback_bp = Blueprint('feedback_bp', __name__) #新增回饋表單 @feedback_bp.route('/create/form') def feedback_create_form(): return render_template('feedback_create_form.html') </pre>	

```
@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")

        # 將回饋資料存入資料庫
        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>
        """

#回饋紀錄
@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)

#管理者回饋紀錄
@admin_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()

    # 查詢所有回饋的總數量
    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

    # 查詢特定頁面的回饋資料，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)

#回覆回饋表單
@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)

#回覆回饋
@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	services\user\app.py
用途	用戶個人資料相關功能
部分程式碼	
<pre> # 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(['png', 'jpg', 'jpeg', 'gif']) # 定義函式來檢查文件擴展名 </pre>	


```

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	services\manager\app.py
用途	管理者相關功能
部分程式碼	
<pre> @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') # 處理測試資料，接收成清單 test_inputs = data.getlist('test_input[]') # 傳回的資料會是清單 </pre>	

```

test_outputs = data.getlist('test_output[]')

# 檔案儲存路徑，基於專案根目錄
base_path = os.path.join(current_app.root_path, 'source', 'testdata')

# 建立新題號的資料夾
new_problem_folder = os.path.join(base_path, f'{new_no:04d}')
if not os.path.exists(new_problem_folder):
    os.makedirs(new_problem_folder)

# 儲存測試資料為檔案
for idx, (test_inp, test_out) in enumerate(zip(test_inputs, test_outputs),
start=1):
    # 檔案名稱格式
    input_filename = os.path.join(base_path,
f'{new_no:04d}_{idx:02d}.in')
    output_filename = os.path.join(base_path,
f'{new_no:04d}_{idx:02d}.out')

    # 將資料寫入檔案，加入具體的錯誤處理
    try:
        with open(input_filename, 'w', encoding='utf-8') as f_in:
            f_in.write(test_inp)

        with open(output_filename, 'w', encoding='utf-8') as f_out:
            f_out.write(test_out)

    except Exception as file_error:
        print(f'Error writing file {input_filename} or {output_filename}:
{file_error}")

    # 此處的 SQL 操作暫時被註解掉，如果需要，可以取消註解
    # 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 Exception as e:
    print(f'Error in add_problem: {e}") # 具體的錯誤資訊

```

```

        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	services\contest\app.py
用途	比賽相關功能
部分程式碼	
<pre> #contest create form @contest_bp.route('/create/form') def contest_create_form(): # 檢查使用者是否已登入 user_id = session.get('User_id') if not user_id: # 如果未登入，渲染 login.html 並顯示提示訊息 #flash('請先登入以參加比賽', 'warning') # 可以顯示提示訊息 return render_template('login.html') # 直接渲染登入頁面 return render_template('create_contest.html') @contest_bp.route('/join/form') def contest_join(): conn = db.connection() cursor = conn.cursor() current_time = datetime.now() user_id = session.get('User_id') # 獲取 URL 中的篩選條件 state = request.args.get('state', '*') contest_type = request.args.get('contest_type', '*') search = request.args.get('search', "").strip() page = request.args.get('page', 1, type=int) per_page = 10 offset = (page - 1) * per_page # 獲取總比賽數的查詢條件 total_contests_query = "SELECT COUNT(*) FROM contest" where_clause = [] params = [] # 添加類型過濾條件 if contest_type != '*' and contest_type: where_clause.append("type = %s") params.append(contest_type) # 添加狀態過濾條件 if state == 'not_started': where_clause.append("start_date > %s") params.append(current_time) </pre>	

```

elif state == 'ongoing':
    where_clause.append("start_date <= %s AND end_date > %s")
    params.extend([current_time, current_time])
elif state == 'finished':
    where_clause.append("end_date <= %s")
    params.append(current_time)

# 添加搜尋條件
if search:
    where_clause.append("contest_name LIKE %s")
    params.append(f"%{search}%")

# 構建完整的 SQL 查詢
if where_clause:
    total_contests_query += " WHERE " + " AND ".join(where_clause)

cursor.execute(total_contests_query, tuple(params))
total_contests = cursor.fetchone()[0]

# 查詢比賽資料
contest_query = """
SELECT contest_id, contest_name, start_date, end_date, description, type
FROM contest
"""
if where_clause:
    contest_query += " WHERE " + " AND ".join(where_clause)
contest_query += " ORDER BY contest_id DESC LIMIT %s OFFSET %s"
cursor.execute(contest_query, tuple(params + [per_page, offset]))
contests = cursor.fetchall()

# 查詢每個比賽的參加人數
contest_participants_query = """
SELECT contest_id, COUNT(user_id)
FROM `contest participant`
GROUP BY contest_id
"""
cursor.execute(contest_participants_query)
contest_participants = cursor.fetchall()

# 將比賽 ID 對應到參加人數
contest_participants_dict = {row[0]: row[1] for row in contest_participants}

# 添加參加狀態到 contests 中
contests_with_status = []
for contest in contests:

```

```

contest_id = contest[0]
participant_count = contest_participants_dict.get(contest_id, 0)

start_date = contest[2]
end_date = contest[3]

remaining_days = (end_date - current_time).days
remaining_seconds = (end_date - current_time).seconds

if start_date > current_time:
    countdown_days = (start_date - current_time).days
    countdown_hours = (start_date - current_time).seconds // 3600
    if countdown_days >= 1:
        time_status = f"倒數 {countdown_days} 天開始"
    elif countdown_days < 1:
        if countdown_hours >= 1:
            time_status = f"倒數 {countdown_hours} 小時開始"
        else:
            time_status = "即將開始"
elif remaining_days > 0 or (remaining_days == 0 and remaining_seconds > 0):
    if remaining_days == 0:
        remaining_hours = (end_date - current_time).seconds // 3600
        time_status = f"比賽將在 {remaining_hours} 小時後結束"
    else:
        time_status = f"剩餘天數: {remaining_days} 天"
else:
    time_status = "比賽已結束"

if user_id:
    cursor.execute("SELECT contest_id FROM `contest participant` WHERE
user_id = %s", (user_id,))
    joined_contests = set([row[0] for row in cursor.fetchall()])
    status = "joined" if contest_id in joined_contests else "not_joined"
else:
    status = "not_joined"

contests_with_status.append((*contest, status, participant_count, time_status))

no_contest_message = None
if not contests:
    no_contest_message = "沒有符合條件的比賽"

total_pages = (total_contests + per_page - 1) // per_page

conn.close()

```



```

return render_template('join_contest_form.html', contests=contests_with_status,
                        page=page, total_pages=total_pages,
                        state=state, contest_type=contest_type, search=search,
current_time=current_time,
                        no_contest_message=no_contest_message)

@contest_bp.route('/join', methods=["POST"])
def join_contest():
    contest_id = request.form['contest_id']
    input_password = request.form.get('contest_password')

    # 請確保 session 中正確設定了使用者 ID
    user_id = session.get('User_id')

    if not user_id:
        # 用戶未登錄，返回狀態碼和訊息
        return jsonify({"error": "请先登录", "code": "not_logged_in"}), 403

    conn = db.connection()
    cursor = conn.cursor()

    cursor.execute("SELECT type, password FROM contest WHERE contest_id = %s",
(contest_id,))
    contest = cursor.fetchone()

    if not contest:
        conn.close()
        return jsonify({"error": "比賽不存在"}), 404

    contest_type, contest_password = contest

    if contest_type == 'private':
        if not input_password or input_password != contest_password:
            return jsonify({"error": "比賽密碼錯誤"}), 403

    cursor.execute("SELECT 1 FROM `contest participant` WHERE contest_id = %s
AND user_id = %s", (contest_id, user_id))
    already_joined = cursor.fetchone()

    if not already_joined:
        cursor.execute("INSERT INTO `contest participant` (contest_id, user_id)
VALUES (%s, %s)", (contest_id, user_id))
        conn.commit()

```

```

conn.close()

return jsonify({"redirect_url": url_for('contest_bp.contest_info',
contest_id=contest_id)})

@contest_bp.route('/contest/<contest_id>', methods=["GET"])
def contest_info(contest_id):
    user_id = session['User_id']

    conn = db.connection()
    cursor = conn.cursor()

    # 檢查該使用者是否有參加該比賽
    cursor.execute("SELECT 1 FROM `contest participant` WHERE contest_id = %s
AND user_id = %s", (contest_id, user_id))
    joined = cursor.fetchone()

    if not joined:
        conn.close()
        return "你未參加該比賽，無法查看相關資訊", 403 # 如果使用者未參加該
比賽，返回 403 錯誤

    # 查詢比賽名稱、開始時間和結束時間
    cursor.execute("SELECT contest_name, start_date, end_date FROM contest
WHERE contest_id = %s", (contest_id,))
    result = cursor.fetchone()

    if result is None:
        conn.close()
        return "比賽不存在", 404 # 如果查不到數據，回傳錯誤訊息

    contest_name = result[0]
    start_time = result[1]
    end_time = result[2]

    # 查詢與該比賽相關的所有題目及是否完成
    cursor.execute("""
        SELECT p.problem_id, p.title, p.difficulty,
            CASE
                WHEN EXISTS (SELECT 1 FROM `contest submission` cs
                    WHERE cs.problem_id = p.problem_id AND
cs.contest_id = %s
                    AND cs.user_id = %s AND cs.is_finish = 'Accepted')
                THEN 1
            ELSE 0
    """)

```

```

        END AS is_finished
    FROM `contest problem` cp
    JOIN `problem` p ON cp.problem_id = p.problem_id
    WHERE cp.contest_id = %s
    """, (contest_id, user_id, contest_id))
    problems = cursor.fetchall()

# Contest Rank 查詢
data = db.get_data(f"
    SELECT
        cs.contest_id,
        u.user_name,
        u.image,
        SUM(CASE WHEN cs.is_finish = 'Accepted' THEN 1 ELSE 0 END) AS 正
確答題數,
        CONCAT(ROUND(SUM(CASE WHEN cs.is_finish = 'Accepted' THEN 1
ELSE 0 END) / COUNT(*) * 100, 2), '%') AS 答題正確率,
        ROUND(AVG(cs.run_time), 2) AS 平均執行時間,
        ROUND(AVG(cs.memory), 2) AS 平均使用記憶體
    FROM `113-CodeAlchemy`.`contest submission` cs
    JOIN `113-CodeAlchemy`.`user` u ON cs.user_id = u.user_id
    WHERE cs.contest_id = {contest_id} -- 直接將 contest_id 嵌入查詢語句中
    GROUP BY cs.contest_id, u.user_name, u.image
    ORDER BY 正確答題數 DESC, 答題正確率 DESC, 平均執行時間, 平均
使用記憶體;
    ")

    conn.close()

# 傳遞 contest_id 和 data 給模板
    return render_template('contest_joined.html', contest_name=contest_name,
start_time=start_time, end_time=end_time, problems=problems, contest_id=contest_id,
data=data)

@contest_bp.route('/get_problems')
def get_problems():
    # 取得 URL 參數中的 page 和 per_page 變量，如果沒有則分別預設為 1 和 10
    page = int(request.args.get('page', 1))
    per_page = int(request.args.get('per_page', 10))

    # 建立到資料庫的連接
    conn = db.connection()
    cur = conn.cursor()

```

```

# 計算應該跳過的記錄數
offset = (page - 1) * per_page
# 執行分页查询
cur.execute("SELECT problem_id, title, content, difficulty FROM problem LIMIT
%s OFFSET %s;", (per_page, offset))
problems_data = cur.fetchall()

# 取得資料庫中題目的總數以計算頁數
cur.execute("SELECT COUNT(*) FROM problem;")
total_problems = cur.fetchone()[0]
total_pages = (total_problems + per_page - 1) // per_page # 計算總分頁數

# 關閉連接
cur.close()
conn.close()

# 返回查詢結果給前端，附加分页信息
return jsonify({
    'data': problems_data,
    'total': total_problems,
    'page': page,
    'total_pages': total_pages,
    'per_page': per_page
})

# 路由：處理創建比賽的 POST 請求
@contest_bp.route('/create', methods=['POST'])
def create_contest():
    connection = None
    try:
        # 從表單取得數據
        contest_name = request.form.get('contest_name')
        start_date = request.form.get('startTime')
        end_date = request.form.get('endTime')
        description = request.form.get('description')
        contest_type = request.form.get('state') # 修改此行來獲取下拉選單的值
        contest_password = None # 預設密碼為 None

        # 如果比賽類型為私人，則獲取密碼
        if contest_type == "private":
            contest_password = request.form.get('contest_password')
            if not contest_password:

```

```

        raise ValueError("私人比賽必須設定密碼")

# 列印接收到的資料以調試
print("Received contest_name:", contest_name)
print("Received start_date:", start_date)
print("Received end_date:", end_date)
print("Received description:", description)
print("Received type:", contest_type) # 修改此行以匹配 contest_type

# 確保所有必要的欄位都被正確接收
if not contest_name or not start_date or not end_date or not description or not
contest_type:
    raise ValueError("All contest fields must be provided")

# 從表單資料中解析題目 ID
contest_data = request.form.to_dict()
problem_ids = contest_data.get('problem_ids', "").split(',')

if not problem_ids or " " in problem_ids:
    raise ValueError("No problem IDs provided")

print("Received problem_ids:", problem_ids)

# 資料庫連線
connection = db.connection()

# 新增到 contest 資料表
cursor = connection.cursor()
cursor.execute("INSERT INTO contest (contest_name, start_date, end_date,
description, type, password) VALUES (%s, %s, %s, %s, %s, %s)",
               (contest_name, start_date, end_date, description, contest_type,
contest_password))

# 取得新建立的 contest_id
contest_id = cursor.lastrowid

# 循環新增題目到 contest problem 表
for problem_id in problem_ids:
    connect_contest_and_problem(connection, contest_id, problem_id)

connection.commit()

# 比賽建立成功後渲染 create_contest_success.html
return render_template('create_contest_success.html')
except Exception as e:

```

```

        print("Error occurred:", e)
        return jsonify({"status": "error", "message": str(e)})
    finally:
        # 確保關閉資料庫連接
        if connection:
            connection.close()

# 輔助函數：將 contest_id 和 problem_id 關聯起來
def connect_contest_and_problem(connection, contest_id, problem_id):
    cursor = connection.cursor()

    # 插入 contest_id 和 problem_id 到 contest problem 表中
    cursor.execute(
        "INSERT INTO `contest problem` (contest_id, problem_id) VALUES (%s, %s);",
        (contest_id, problem_id)
    )

    cursor.close()

```

路徑	utils\dolos.py
用途	程式碼分析功能
部分程式碼	
<pre> def create_zip(problem_id): pattern = f'*_{problem_id}*' files = glob.glob(os.path.join('source', pattern)) # 指定輸出 Zip 檔案的路徑 output_zip_path = f'dolos/{uuid.uuid4()}_{problem_id}.zip' # 確保輸出的目錄存在 os.makedirs(os.path.dirname(output_zip_path), exist_ok=True) # 創建一個新的 Zip 檔案 with zipfile.ZipFile(output_zip_path, 'w') as zipf: for file in files: zipf.write(file, os.path.basename(file)) return (os.path.basename(output_zip_path), output_zip_path) def submit_to_dolos(name, zipfile_path): """ Submit a ZIP-file to the Dolos API for plagiarism detection and return the URL where the resulting HTML report can be found. </pre>	

```

"""
# 提交 ZIP 檔案到 Dolos API
with open(zipfile_path, 'rb') as zip_file:
    response = requests.post(
        'http://140.131.114.141:3000/reports',
        files={'dataset[zipfile]': zip_file},
        data={'dataset[name]': name}
    )

# 解析 JSON 響應
json_response = response.json()

# 刪除已提交的 ZIP 檔案
try:
    os.remove(zipfile_path)
    print(f'已刪除檔案: {zipfile_path}')
except Exception as e:
    print(f'刪除檔案 {zipfile_path} 時發生錯誤: {e}')

# 返回 HTML 報告的 URL
return json_response["html_url"]

```

路徑	crawler\add_problem.py
用途	管理員上傳新的題目
部分程式碼	
<pre> 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' </pre>	

```

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	
<pre> 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': </pre>	

```

        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

```

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。

2. 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。

3. 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。

4. 回饋

4-1 新增回饋：使用者是否可以新增意見回饋。

4-2 回饋結果：使用者是否可以查看回饋結果。

5. 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。

6. 回饋管理

6-1 回饋結果：確認管理者是否能將結果回饋給使用者。

6-2 回饋內容：確認管理者是否能收到使用者回饋的內容。

7. 使用者管理

7-1 管理者修改使用者的權限。

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 新增回饋測試

功能編號	4-1	功能名稱	新增回饋
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"意見回饋" 4.輸入回饋內容 5.點擊"提交回饋"		
預期結果	提交成功		
執行結果	送出成功		

▼表 10-2-4-2 回饋結果測試

功能編號	4-2	功能名稱	回饋結果
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"意見回饋" 4.點擊"回饋紀錄"		
預期結果	成功查看回饋結果		
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 10-2-6-1 回饋結果測試

功能編號	6-1	功能名稱	回饋結果
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"意見回饋"		
預期結果	成功查看回饋結果		
執行結果	測試成功		

▼表 10-2-6-2 回饋內容測試

功能編號	6-2	功能名稱	回饋內容
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"意見回饋" 4.選擇要回覆的回饋，點擊"回覆" 5.輸入回饋內容 6.點擊"回覆"		
預期結果	回覆成功		
執行結果	送出成功		

▼表 10-2-7 使用者管理測試

功能編號	7-1	功能名稱	使用者管理
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"管理使用者" 4.選擇要更改的使用者更改權限		
預期結果	更改成功		
執行結果	更改成功		

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
系統使用套件	Flask v3.0.2 Flask-mail v0.9.0 Flask-login v0.6.3 PyMySQL v1.1.0 selenium v4.19.0 webdriver_manager v4.0.1 pandas v2.2.1 Authlib v1.3.0 psutil v5.9.8 bs4 v0.0.1 flask_paginate v2024.4.12 psutil v5.9.8 requests 2.31.0 lxml 5.2.2 html5lib 1.1

第十二章 使用手冊

▼表 12-1 進入 CodeAlchemy

畫面名稱

說明

進入 CodeAlchemy

進入 CodeAlchemy 主頁。

CodeAlchemy

題目 排行榜 競賽

Login or Sign-up

狀態

OnlineJudge

難度

標籤

名稱

通過	OJ	編號	名稱	難度	標籤	通過率	上傳日期
	CodeAlchemy	CAOJ-1001	Hello GAWA!!	easy	N/A	None	2024-05-20
	TIOJ	TIOJ-1001	Hello World	easy	N/A	76.85%	2024-05-20
	TIOJ	TIOJ-1002	A+B Problem	easy	N/A	40.00%	2024-05-20
	TIOJ	TIOJ-1003	切義大利餅問題	easy	CKcon 2006-CKcon	0.00%	2024-05-20
	TIOJ	TIOJ-1004	猶太人敢死隊問題	easy	CKcon 2006-CKcon math brute-force	0.00%	2024-05-20
	TIOJ	TIOJ-1005	圓周率問題	hard	CKcon 2006-CKcon math	100.00%	2024-05-20
	TIOJ	TIOJ-1006	大數除法	hard	CKcon 2006-CKcon math	None	2024-05-20
	TIOJ	TIOJ-1007	燈泡問題	normal	CKcon 2006-CKcon CKsim 2015-CKsim-3rd DP	100.00%	2024-05-20
	TIOJ	TIOJ-1008	量杯問題	hard	CKcon 2006-CKcon BFS	0.00%	2024-05-20
	TIOJ	TIOJ-1009	In No Time!	normal	2006-CKsim-1st CKsim implementation	100.00%	2024-05-20
	TIOJ	TIOJ-1010	Prefix and Postfix	normal	2006-CKsim-1st CKsim string brute-force	None	2024-05-20
	TIOJ	TIOJ-1011	Edit Distance In Numbers	normal	2006-CKsim-1st CKsim math strings	66.67%	2024-05-20
	TIOJ	TIOJ-1012	Paile	hard	2006-CKsim-1st CKsim-1st data structures stack	100.00%	2024-05-20

▼表 12-2 使用者登入

畫面名稱	使用者登入
說明	點擊"Login or Sign-up"進入使用者登入頁面，同時可以使用 Google 帳號登入 CodeAlchemy。

 CodeAlchemy 題目 排行榜 競賽 ▾

Login or Sign-up

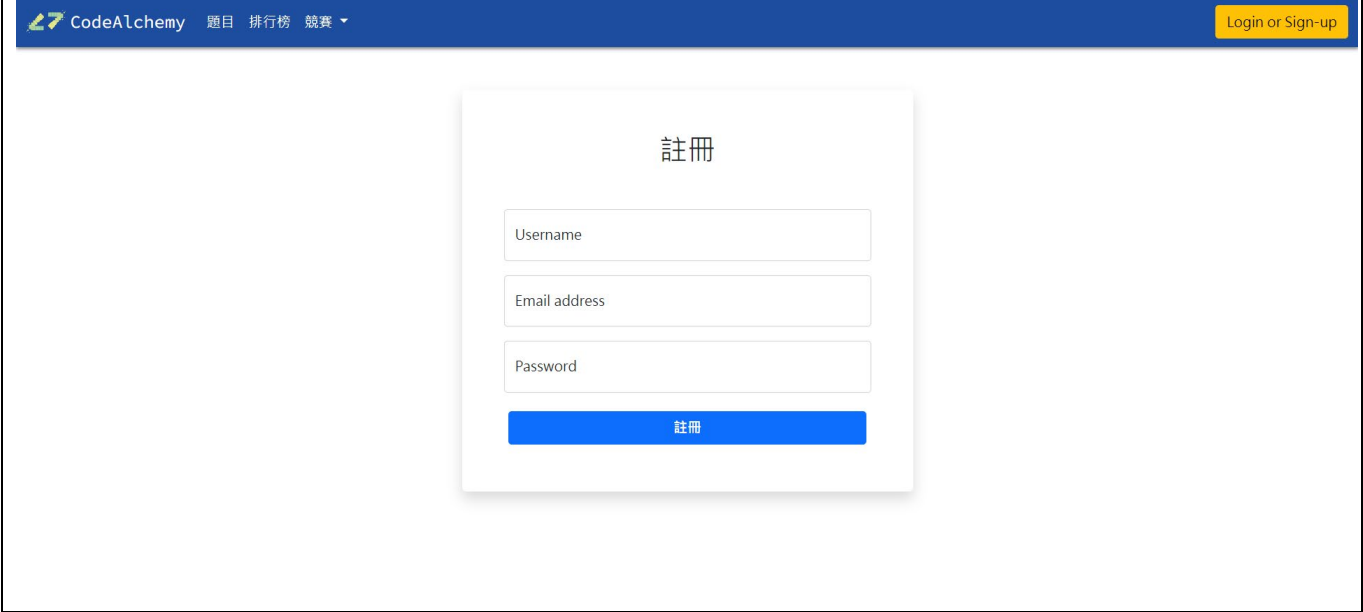
登入或註冊

Email address

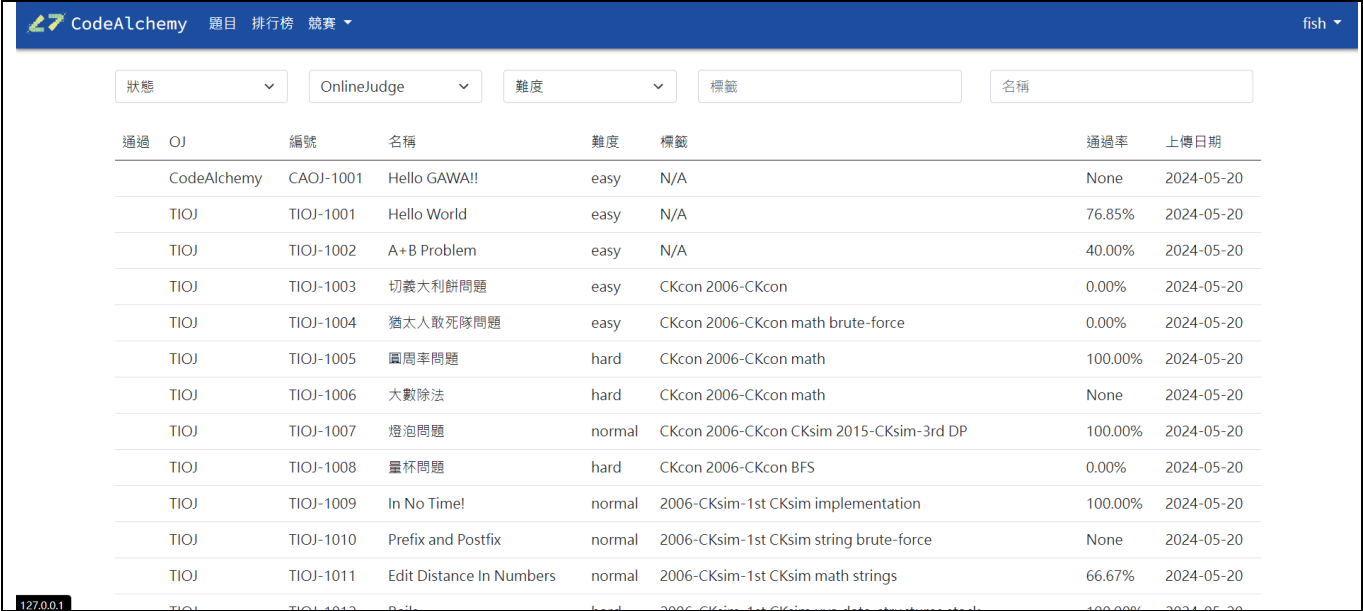
送出

 使用Google登入

▼表 12-3 使用者註冊

畫面名稱	使用者註冊
說明	若登入時的 Email 是沒註冊過的，則會進入註冊頁面。
	

▼表 12-4 使用者查看題目列表

畫面名稱	使用者查看題目列表
說明	登入後即可選擇題目做題。
	







▼表 12-5 使用者上傳題目

畫面名稱	使用者上傳題目
說明	選擇題目後會進入題目內容和上傳的頁面，點擊"上傳"即可將寫好的程式碼上傳。

▼表 12-6 程式碼分析

畫面名稱	程式碼分析
說明	點擊"程式碼分析"後會進入 DOLOS 頁面，會將該題的分析結果顯示。

▼表 12-7 作答情況

畫面名稱	作答情況						
說明	點擊"作答情況"會顯示該題目有哪些使用者進行了作答。						
<div>CodeAlchemy 題目 排行榜 競賽 fish</div>							
作答編號	用戶編號	用戶姓名	作答結果	程式語言	執行時間	記憶體	上傳日期
161	3	 enyu	Accepted	Python	25.6MS	17.25MB	2024-10-11 18:46:10
160	3	 enyu	Accepted	Python	25.2MS	17.25MB	2024-10-11 18:45:46
159	3	 enyu	Accepted	Python	25.4MS	17.25MB	2024-10-11 18:39:47
158	3	 enyu	Accepted	Python	25.6MS	17.25MB	2024-10-11 18:37:07
157	3	 enyu	Accepted	Python	25.8MS	16.29MB	2024-10-11 03:01:04
156	3	 enyu	Rejected	Python	0.0MS	0.0MB	2024-10-11 03:00:40
155	3	 enyu	Accepted	Python	25.7MS	16.29MB	2024-10-11 03:00:20
154	3	 enyu	Accepted	Python	25.6MS	16.29MB	2024-10-11 02:58:22
153	3	 enyu	Accepted	Python	25.4MS	16.29MB	2024-10-11 02:57:55
152	3	 enyu	Accepted	Python	25.7MS	16.29MB	2024-10-11 02:57:37
151	3	 enyu	Accepted	Python	25.7MS	17.25MB	2024-10-11 02:57:16
150	3	 enyu	Accepted	Python	25.4MS	17.25MB	2024-10-11 02:56:30
149	3	 enyu	Rejected	C++	0.0MS	0.0MB	2024-10-11 02:42:47

▼表 12-8 使用者查看排行榜

畫面名稱	使用者查看排行榜				
說明	點擊"排行榜"將會顯示全體使用者解題排行。				

 CodeAlchemy

題目 排行榜 競賽 ▾

fish ▾

名次	用戶名稱	正確答題數	正確率	平均執行時間	平均使用記憶體
1	 enyu	48	70.59%	19.14 ms	12.96 MB
2	 yen	36	78.26%	20.21 ms	13.19 MB
3	 BoYoOuO	7	58.33%	39.73 ms	28.06 MB
4	 fyc	5	55.56%	49.47 ms	27.53 MB
5	 Enyu	5	83.33%	40.74 ms	25.08 MB
6	 fanyucian	4	80.00%	30.36 ms	24.26 MB
7	 fish0.0	3	60.00%	38.32 ms	27.82 MB
8	 Alicia	1	33.33%	36.75 ms	21.44 MB
9	 esn	1	50.00%	53.86 ms	23.74 MB

▼表 12-9 使用者創建比賽

畫面名稱	使用者創建比賽
說明	使用者可以透過創建比賽來為其他使用者進行比賽。

CodeAlChemy 題目 排行榜 競賽 fish

創建比賽

比賽名稱:

請輸入比賽名稱

比賽說明:

請輸入比賽說明

比賽類型:

公開

開始時間:

年 / 月 / 日 -- --:--

結束時間:

年 / 月 / 日 -- --:--

題目列表:

#	Name	Difficulty	Remove
+ 新增題目			

取消

創建比賽

▼表 12-10 新增題目

畫面名稱	新增題目
說明	比賽題目可透過系統內所有題目進行挑選。

CodeAlChemy 題目 排行榜 競賽 fish

題目編號	題目名稱	題目說明	難易度	Action
CAOJ-1001	Hello GAWAI!	請你輸出"Hello Tmt World XD!"	easy	Add
TIOJ-1001	Hello World	請你輸出"Hello Tmt World XD!"	easy	Add
TIOJ-1002	A+B Problem	請計算 $Sa + bS$ 。	easy	Add
TIOJ-1003	切嘉大利肝問題	電腦程式語言的稱呼中，這個函數可說是較特殊的一種，因函數可以呼叫函數本身自己。 請設計一個程式，用這個函數解決下面兩小題(Problem 1003, Problem 1004 more)	easy	Add
TIOJ-1004	國大人數扣除問題	於某一定紀時，國大人受制於羅馬帝國，一批美國志士反抗羅馬帝國，最後僅剩下41人受困於山南，這41人不想受辱於羅馬帝國，又不想自殺，因此決定重成一區亂黨，由1號開始每2號殺掉，3號每4號殺掉，依此 more	easy	Add
TIOJ-1005	圖周車問題	與著一個大藥的師Robert A.J. Matthews根據夜空中劃過天際的星量的位置，讓人驚訝的推論出關於 $\$ \pi i S$ (星周車) 的準確度。當然，該黨扯到數論的複雜及應用，在此，我們沒有後空，但 more	hard	Add
TIOJ-1006	大數除法	請將兩個50位數以內的整數數字相除，列印出結果。	hard	Add
TIOJ-1007	燈泡問題	我們有一些不同的燈泡，燈罩都有一條燈絲，但品質都不一樣好，如果持續地讓燈泡亮著的話，燈絲這一會兒就燒斷了，因此必須有持續電源切斷，讓燈絲降溫。 假設一個燈泡最多可以持續點 more	normal	Add
TIOJ-1008	量杯問題	這裡有各種不同容量的量杯可以取水，可是有時候所有量杯的容量都不是剛好我們想要的，因此我們要用些技巧來量出我們所需要的水量。 舉例來說，這裡有一個 3 公升和一個 5 公升的 more	hard	Add
TIOJ-1009	In No Time!	考試時間所剩不多，你能夠把還剩下時間完成所有題目嗎？給你現在時間以及考試截止時間，請你判斷還剩下多少時間可以做題目。	normal	Add

<< 1 2 3 >>

▼表 12-11 使用者加入比賽

畫面名稱	使用者加入比賽
說明	使用者可以透過參加比賽方式提升程式能力。
	

▼表 12-12 私人比賽

畫面名稱	私人比賽
說明	加入私人比賽需要相應的密碼才可加入。
	

▼表 12-13 比賽題目

畫面名稱	比賽題目
說明	挑選想加入的比賽後會出現比賽裡需要解題的題目，點擊題目即可作答。


題目 排行榜 競賽 ▾
fish ▾

開始時間: 2024-10-11 00:42:00
public 8777
結束時間: 2024-10-15 00:42:00
比賽進行中!

題目 排名

#	題目編號	題目名稱	難度	finish
1	TIOJ-1001	Hello World	easy	
2	TIOJ-1026	惡猿果實	normal	
3	TIOJ-1027	高精度開平方根	hard	

▼表 12-14 比賽排名

畫面名稱	比賽排名
說明	使用者可以透過排名查看比賽名次。


題目 排行榜 競賽 ▾
fish ▾

開始時間: 2024-10-11 00:42:00
public 8777
結束時間: 2024-10-15 00:42:00
比賽進行中!

題目 排名

名次	用戶名稱	正確答題數	正確率	平均執行時間	平均使用記憶體
1	 yen	1	100.00%	25.3ms	17.25MB

▼表 12-15 使用者個人檔案

畫面名稱	使用者個人檔案
說明	點擊右上使用者名稱會進入個人資料，除了使用者 ID 以外都可以做修改。



▼表 12-16 使用者個人解題紀錄

畫面名稱	使用者個人解題紀錄
說明	點擊"個人解題紀錄"即可看到使用者目前作答了哪些題目。



題號	語言	執行時間	花費記憶體	上傳時間	結果
TIOJ-1011	c++	38.15	32.98	2024-06-01 15:59:49	通過
TIOJ-1011	python	21.89	37.01	2024-06-01 16:00:03	未通過
TIOJ-1001	python	35.63	28.57	2024-06-01 16:00:04	未通過
TIOJ-1021	c++	50.27	21.91	2024-06-01 16:00:14	通過
TIOJ-1011	python	45.64	18.65	2024-06-01 16:00:47	通過

▼表 12-17 使用者收藏紀錄

畫面名稱	使用者收藏紀錄
說明	點擊"收藏紀錄"即可看到使用者收藏的題目。



▼表 12-18 使用者回饋

畫面名稱	使用者回饋
說明	使用者可以透過回饋方式提出系統需要修改的地方。



▼表 12-19 回饋紀錄

畫面名稱	回饋紀錄						
說明	回饋後即可透過紀錄來查看回饋進度和結果。						
<div>CodeAlchemy 題目 排行榜 競賽 fish0.0</div> <div><table><thead><tr><th>反饋內容</th><th>提交時間</th><th>回覆</th></tr></thead><tbody><tr><td>上傳跑太慢了</td><td>2024-10-12 21:56:41</td><td>None</td></tr></tbody></table></div>		反饋內容	提交時間	回覆	上傳跑太慢了	2024-10-12 21:56:41	None
反饋內容	提交時間	回覆					
上傳跑太慢了	2024-10-12 21:56:41	None					

▼表 12-20 管理者題目列表

畫面名稱	管理者題目列表
說明	管理者可查看所有題目。

CodeAlchemy

題目

排行榜

競賽

fish0.0

狀態

OnlineJudge

難度

標籤

名稱

通過	OJ	編號	名稱	難度	標籤	通過率	上傳日期	修改	刪除
	CodeAlchemy	CAOJ-1001	Hello GAWA!!	easy	N/A	None	2024-05-20		
! 未通過	TIOJ	TIOJ-1001	Hello World	easy	N/A	76.85%	2024-05-20		
	TIOJ	TIOJ-1002	A+B Problem	easy	N/A	40.00%	2024-05-20		
	TIOJ	TIOJ-1003	切義大利餅問題	easy	CKcon 2006-CKcon	0.00%	2024-05-20		
	TIOJ	TIOJ-1004	猶太人敢死隊問題	easy	CKcon 2006-CKcon math brute-force	0.00%	2024-05-20		
	TIOJ	TIOJ-1005	圓周率問題	hard	CKcon 2006-CKcon math	100.00%	2024-05-20		
	TIOJ	TIOJ-1006	大數除法	hard	CKcon 2006-CKcon math	None	2024-05-20		
	TIOJ	TIOJ-1007	燈泡問題	normal	CKcon 2006-CKcon CKsim 2015-CKsim-3rd DP	100.00%	2024-05-20		
	TIOJ	TIOJ-1008	量杯問題	hard	CKcon 2006-CKcon BFS	0.00%	2024-05-20		

▼表 12-21 管理者修改題目

畫面名稱	管理者修改題目
說明	管理者可挑選需要修改的題目，並對題目內容做修正。
	<p>題目標題:</p> <input type="text" value="Hello World"/> <p>題目說明:</p> <input type="text" value="<p>請仔細閱讀「Hello Test World XDI」</p>"/> <p>輸入說明:</p> <input type="text" value="<p>這裡沒有input。</p>"/> <p>輸出說明:</p> <input type="text" value="<p>Hello Test World XDI</p>"/> <p>範例輸入:</p> <input type="text" value="<pre>./No Input</pre>"/> <p>範例輸出:</p> <input type="text" value="<pre>Hello Test World XDI</pre>"/> <p>測試資料輸入:</p> <input type="text"/> <p>測試資料輸出:</p> <input type="text"/> <p>難度</p> <input type="text" value="Easy"/> <p>標籤</p> <input type="text" value="N/A"/> <p>影片ID</p> <input type="text" value="QEqIQNvy8"/> <p>新增</p>

▼表 12-22 管理者新增題目

畫面名稱	管理者新增題目
說明	管理者可新增更多題目來提升使用者程式能力。
	<p>CodeAlchemy 題目 排行榜 競賽 fish0.0</p> <p>題目標題:</p> <input type="text"/> <p>題目說明:</p> <input type="text"/> <p>輸入說明:</p> <input type="text"/> <p>輸出說明:</p> <input type="text"/> <p>範例輸入:</p> <input type="text"/> <p>範例輸出:</p> <input type="text"/>

▼表 12-23 管理意見回饋

畫面名稱	管理意見回饋
說明	管理者可透過回饋來查看系統需改進的地方。
	

▼表 12-24 回饋回覆

畫面名稱	回饋回覆
說明	管理者可回覆使用者回饋的問題。
	

▼表 12-25 管理使用者

畫面名稱	回饋回覆
說明	管理者可更改使用者權限。

 CodeAlchemy

題目 排行榜 競賽 ▾

fanyucian ▾

使用者ID	使用者名稱	權限
3	enyu	管理者 ▾
4	yen	管理者 ▾
5	fish0.0	管理者 ▾
6	fanyucian	管理者 ▾
7	esn	封禁 ▾
16	fyc	使用者 ▾
17	BoYoOuO	使用者 ▾
18	Alicia	使用者 ▾
19	Enyu	使用者 ▾
20	codealchemyproject	管理者 ▾

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

透過 CodeAlchemy 的開發歷程，我深刻體會到協作與持續學習是創建一個成功的學習平台不可或缺的要素。這個平台的誕生，凝聚了無數團隊成員的智慧與心血，在不斷的討論與反饋中逐步完善。我從中學習到如何深入了解使用者的需求，並在創新與實用性之間取得平衡。未來，我希望能將 CodeAlchemy 打造成一個陪伴無數程式學習者成長的良師益友，一同探索程式世界的奧秘。同時，我們也將持續優化平台，使其成為一個開放、包容且高效的學習環境，為廣大學習者提供便利與啟發。

在開發 CodeAlchemy 時，我們將題庫與 Dolos 系統進行深度整合，旨在打造一個不僅能幫助使用者掌握解題技巧，更能引導他們深入探索不同解法背後邏輯的學習平台。程式設計不只是單純地解決問題，更重要的是理解解題的思維過程，並學會用多種角度思考問題。在整合過程中，我們面臨了如何將 Dolos 系統與平台無縫銜接的挑戰。特別是透過 Dolos 的 API，實現程式碼相似度檢測與特徵分類的功能，是平台建設的一大亮點。這種深度整合不僅能自動化分析使用者的解題過程，還能為他們呈現同一問題的各種解法，提供更豐富多元的學習體驗。

10956030 林妤玟

一開始，我們發現許多現有的學習平台存在一些問題，例如學習資源分散及缺乏針對使用者的即時反饋。這些問題常常讓學習者感到迷茫或挫折，我們深入研究了學習平台，並找到了一套能夠個性化學習路徑的系統，幫助學習者根據自己的進度進行學習。此外，系統中的競賽功能不僅能夠激發學習者的興趣，還能透過即時排名讓使用者看到自己的成長進步，這些設計的核心理念，是希望在 CodeAlchemy 平台上學習中能注入更多互動性和挑戰性，讓學習變得更加生動。

在系統開發過程中，團隊合作的重要性也變得格外突出。每個階段，我們都需要進行需求分析、功能設計、寫程式、測試以及改進，這些工作都依賴於團隊成員間的分工與溝通。面對不同的意見和技術難題，我們常常需要共同討論，最終找到最適合的解決方案。這樣的合作經驗，讓我學會如何更有效地與他人共同合作，也

理解到專題成功的關鍵在於每個成員的共同努力。

10956044 顏子綺

這次為期一年的專題製作讓我學到很多寶貴的經驗，從一開始的主題發想，到最後的功能設計與實現，每一步都充滿了挑戰。在開發的過程中，時間管理與團隊成員之間的溝通與協調變得尤為重要，謝謝每位組員在這一年中充分發揮每個人的強項，讓整個團隊朝著共同的目標前進。雖然在開發期間中並不是那麼順利，但也讓我們學會了如何去解決各項問題，並且靈活的調整策略。在這一年裡，不僅在程式方面得到了提升，團隊協作與解決問題的能力也同樣得到了極大的成長，期待這些經歷與學習能成為我未來工作中的重要基礎。

最後謝謝我們還有一個很強大的組員叫 ChatGPT，專題開發初期對於透過與 AI 溝通來撰寫程式的這項技能，對我來說還是個陌生的領域，但隨著專題進行，也越來越發現，AI 不僅能幫助我們解決技術難題，還能提供不同的視角和建議，讓開發過程更加順利。無論是解釋複雜的程式邏輯，還是優化我們的程式碼架構，AI 都能在關鍵時刻提供極具價值的幫助。這次的專題中我也透過 AI 快速學習了很多新領域的知識，這也讓我的技能提升變得更加有效率。

10956051 范雨茜

在這次專題讓我學到了很多東西，也知道了很多自己的不足，我明白了一個好的專題必須要有明確的主題並做好分工，要經常去溝通、協調才能越做越好，每個人都要好自己的工作，不能掉隊，大家都要努力，只要有人的工作沒有完成，可能會讓整組的努力都白費，在這次的專題讓我發現我所學的還遠遠不夠，很多東西我都不懂，必須自己去查資料和摸索，但經過這次專題後讓我成長了很多，也學會了很多東西，相信我以後再有類似的事情時一定能夠做的更好。

透過這次的專題，希望我們的平台可以對喜歡寫程式的人有幫助，讓他們能夠充分練習，幫助他們拓寬思維，學會更多、更高效的解題方式，也歡迎大家對我們的平台提出建議，讓我們了解到自己的不足，我們會整合這些意見去改進我們的平台，希望大家會喜歡我們的平台。

第十四章 參考資料

- 簡介(Introduction) · Bootstrap 5 繁體中文文件 - 六角學院
<https://bootstrap5.hexschool.com/docs/5.1/getting-started/introduction/>
- CodeMirror 實現網頁端代碼編輯器
<https://jacychu.medium.com/codemirror-526c359631a7>
- Dolos Official Documentation
<https://dolos.ugent.be/docs/>
- ZeroJudge
<https://github.com/jiangsir/ZeroJudge>
- Flask 整合 Google 登入(Google Sign-In)
<https://myapollo.com.tw/blog/integrating-google-sign-in-into-flask-app/>
- TIOJ INFOR ONLINE JUDGE
<https://tioj.ck.tp.edu.tw/>
- Dolos 安裝
<https://hackmd.io/@10946009/S1gH0mvQ0>
- Gmail API 電子郵件服務設計
<https://hackmd.io/6gDNucH2R3OvY-lQ5q9vmw>
- Flask 實作_ext_09_Flask-Mail_郵件發送
<https://hackmd.io/@shaoeChen/HJiZtEngG/https%3A%2F%2Fhackmd.io%2Fs%2FBytvGKs4M>
- flask-mail 遇到的编码问题
https://blog.csdn.net/weixin_44948269/article/details/131757172
- Python Web Flask — 如何雜湊化使用者密碼？
<https://medium.com/seaniap/python-web-flask-%E5%A6%82%E4%BD%95%E9%9B%9C%E6%B9%8A%E5%8C%96%E4%BD%BF%E7%94%A8%E8%80%85%E5%AF%86%E7%A2%BC-b6dec03c3332>

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。	由於解題程式的寫作受限於語法規範和題目設計，相似度較高的情況時常發生。因此，我們在賽後檢討時，強調的是 Dolos 的程式碼分群功能，幫助參賽者快速了解同一題目的多種解法，深入分析各種解題策略的優缺點。
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。	已修正
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。	已經存在 tag 欄位用於分類及篩選
使用現成套件應說明清楚	已修正
提供分析實例	已新增其他範例
系統目標應更明確聚焦	已修正
主題與目標不明確，有點雜	已修正
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	我們希望運用 Dolos 系統去探討程式碼的結構差異，進一步視覺化多種解法，讓學習者提升實戰能力
建議重新勾勒題目與內容	我們已重新定位系統核心目標為促進程式碼解法的多樣性分析與學習優化，以強化學習者的理解力與創新能力，而非僅限於檢測相似度。