目錄

第一章　前言

1-1 背景簡介

最早的程式語言是在電腦發明之前產生的，當時是用來控制提花織布機及自動演奏鋼琴的動作。在電腦領域已發明了上千不同的程式語言，而且每年仍有新的程式語言誕生。很多程式語言需要用指令方式說明計算的程式，而有些程式語言則屬於宣告式程式設計，說明需要的結果，而不說明如何計算。

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

在過去的學習和實踐過程中，我們曾經花費大量的時間在尋找高質量的學習資源上，並常常感到孤獨和困惑。我們意識到現有的學習平台雖然提供了大量的題目和教材，但傳統的教育方式往往缺乏趣味性和互動性，使學生難以找到與之相符的學習方法；然而對於職場工作者和專業技術人員來說，則是時間和精力的限制成為了學習程式設計的一大阻礙，因此在互動性、個性化學習和社群建設方面還有很大的改進空間。

1-2 動機

我們的動機源於對技術教育的貢獻渴望。CodeAlchemy 使命是打造一個開放、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享知識、交流經驗、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：提供廣泛且高質量的程式設計學習資源，包括教程、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. 促進討論交流：提供一個互動交流的平台，讓學習者都能夠彼此交流學習心得、分享經驗、解決問題，從而促進學習的效果和效率。

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

4. 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。

1-4 預期成果

本專題主要在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質量教材、教程和練習題，並打造互動交流的社群環境，讓學習者能夠和專家與其他技術愛好者進行討論和分享，彼此學習、共同成長。透過技術交流和實踐，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 運營計畫

2-1 可行性分析

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

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

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

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

2-2 商業模式

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

2. 教育與學術市場模式：基於檢查抄襲的功能，廣泛應用於學術課堂。透過測驗來了解學生在編程方面的學習能力，同時預防學生的作弊行為。此外，為支持教學和學術研究活動，以滿足不同學術機構和研究人員的需求。

3. API授權模式：提供應用程式界面（API），供開發人員將程式相似度比對功能集成到他們自己的應用程式中。這意味著開發人員可以根據自己的需求，靈活地將CodeAlchemy的功能整合到他們的應用程式中，從而提高程式碼的品質和可靠性。

4. 企業授權模板：針對企業市場提供特殊的授權模式，允許多位開發人員進行程式碼的品質維護。這種模式有助於提升開發團隊的協作效率，並且透過集中化的管理，實現更友善的版本控制。

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、培訓中心等。初學程式設計、技術面試準備者等。

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

|  |  |
| --- | --- |
| 優勢(Strengths)  1.提供中文題庫，讓看不懂英文的人也能輕易讀懂題目。  2.用戶可以分享知識、交流經驗，增加用戶忠誠度。  3.提供了多樣化的學習資源，各種程式語言的經典類型題目，滿足不同用戶的學習需求。 | 弱勢(Weaknesses)  1.要持續開發和維護一個高質量的教育平台需要大量的技術和資源。  2.無法有效地收集和利用用戶的反饋和建議，無法及時調整和改進平台，可能會影響用戶滿意度。  3.系統無法處理某些程式語言，需要進一步改善。 |
| 機會(Opportunities)  1.可以透過不斷更新，以此來精進平台的精準度和性能 2.分析差異化的競爭策略，提供定制化的解決方案、特色課程、行業專家支持等，來吸引更多的用戶。 | 威脅(Threats)  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.程式設計教育市場可能已經飽和，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。 |

TOWS 分析

|  |  |
| --- | --- |
| 優勢/機會(SO)  1.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。 | 弱勢/機會(WO)  1.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。 |
| 優勢/威脅(ST)  1.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。 | 弱勢/威脅(WT)  1.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。 |

4-2 專案組織與分工

●主要負責人 〇次要負責人 (每一項只能有1位主要負責人，次要負責人最多2位)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 項目/組員 | | 10956010  童柏佑 | 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 | ● |  |  | 〇 | 〇 |
| 伺服器架設 | 〇 | ● |  |  |  |
| 功能1 帳號管理 |  | ● |  |  |  |
| 功能2 題目管理 |  | ● |  |  |  |
| 功能3 比賽管理 | ● |  |  |  |  |
| 功能4 討論群管理 | ● |  |  |  |  |
| 功能5 反饋管理 |  |  |  |  | ● |
|  |  |  |  |  |  |
| 爬蟲 | ● | 〇 |  |  |  |
| 前端開發 | 帳號註冊頁面 |  | ● |  |  |  |
| 問題 |  | ● |  |  |  |
| 系統主畫面 |  | ● |  |  |  |
| 排行榜 |  |  |  |  |  |
| 比賽 |  |  |  |  |  |
| 會員資料 |  |  |  |  |  |
| 程式碼分析畫面 |  |  |  |  |  |
| 討論群 |  |  |  |  |  |
| 反饋 |  |  |  |  |  |
| 美術設計 | UI/ UX |  |  |  |  |  |
| Web/APP介面設計 |  |  |  |  |  |
| 色彩設計 |  |  |  |  |  |
| Logo設計 |  |  |  | ● |  |
| 素材設計 |  |  |  | ● |  |
| 文件撰寫 | 統整 |  |  | ● |  | 〇 |
| 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 |  |  | 〇 |  | ● |
| 第5章 需求模型 |  |  | 〇 | 〇 | ● |
| 第6章 程序或設計模型 |  |  |  | 〇 | ● |
| 第7章 資料或實作模型 |  |  |  | 〇 | ● |
| 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 |  |  |  |  |  |
| 海報製作 |  |  |  |  |  |
| 影片製作 |  |  |  |  |  |

**第五章 需求模型**

**5-1 功能分解圖**

**5-2 需求清單**

|  |  |  |
| --- | --- | --- |
| **編號** | **功能** | **功能敘述** |
| **1** | 程式碼評估 | 使用者可以提交自己的解題程式碼，獲得即時的評估和反饋。 提供自動化的程式碼評測和錯誤檢測，幫助使用者改進解題能力。 |
| **2** | 題目管理 | 使用者可以搜尋、篩選和瀏覽各種程式設計題目。 提供不同的題目分類、標籤和難度等選項，讓使用者能夠快速找到他們感興趣的題目。 |
| **3** | 討論區 | 提供討論區或評論功能，讓使用者可以就特定題目進行討論和交流。 使用者可以發表問題、回答疑問、分享解題思路和獲取反饋。 使用者可以查詢有幾種不同的解題方式。 |
| **4** | 比賽功能 | 平台使用者可以組織和舉辦程式設計競賽活動，並向其他用戶公開宣布比賽的日期、時間、格式和規則。 使用者可以在平台上報名參加感興趣的競賽，參與競賽活動。 |
| **5** | 相似度比對 |  |

**功能性需求：**

**非功能性需求：**

效能需求：確保系統能夠快速、穩定地處理大量的用戶請求和數據操作。 最大限度地優化系統性能，盡量減少頁面加載時間和服務器延遲。

可用性：提供直覺友好的用戶界面，讓使用者能夠輕鬆找到所需要的功能並進行操作。

穩定性：測試系統的穩定性和可靠性，確保在高流量和高負載情況下仍能正常運行。