

參賽隊名：普羅程式

作品名稱：整合式線上教學平台 ProgLearn

1. 作品簡介

(a) 作品目的

本計畫將建立一個名為 ProgLearn 的教學工具，專注於教師導向的教學。該工具將提供課堂教學系統和線上解題系統 (OJ, Online Judge)，前者包含課程影音、互動式講義和課堂練習。

(b) 作品動機與研究問題

在台灣，資訊科技領域受廣泛重視，且程式設計已被列為學校必修課程 [1]。儘管每年有數百萬學生修習程式課程 [2]，但是在基層教育中卻存在許多問題：

(1) 教師在教學與備課上的工作量大：

比如準備線上課程的直播環境、準備課程教材、批改學生作業、上傳考題、課外時間回覆學生的各種問題等 [4]。由於專業教師與學生數量的比例失衡，也讓教師難以管理學生 [3]。

(2) 受技術限制的教學方式：

- 線下課程：資訊教室大多使用資訊教室的廣播與管理系統，讓教師能夠強制監控所有學生的電腦，並在課堂中分享自己的畫面。雖然這種方式能讓學生專注於教師的授課中，但強制控制學生的電腦，學生就無法在上課過程中使用顯示畫面以外的電腦功能，如查詢資料、閱讀過往的課程講義、實作程式碼等，大大限制學生的學習空間。
- 線上課程：教師的教學設備品質，會影響學生學習情緒及上課意願 [5]。此外，線上自主學習，學生若有問題無法獲得即時解答，學習效果可能受限 [4]。

(3) 師生間缺乏互動性：

- 線下課程：資訊教室要實現師生間的雙向互動，大多以麥克風作為訊息傳遞的媒介，但對於有限的麥克風數量、有限的教室空間，增加教師教學的難度。
- 線上課程：教師大多以直播軟體搭配投影片簡報軟體教學，由於缺乏實際互動，難以保持學生的學習熱忱，教師也無法即時注意學生的反應而影響其教學成效 [5]。此外，透過軟體建置的網路平台，並非只能以影音、圖文作為訊息的傳遞方式，比如 youtube 的 Crowd Choice 功能、youtube 直播時的即時問答功能、HackMD 的線上共筆功能、Twitch 實況互動元件等，都是可以被利用的互動性功能。教師應提供好的遠距教學平台與工具，增進師生互動 [4]。

(4) 教學工具種類繁多且功能單一：

- 線下課程：教師在課堂中常使用資訊教室的廣播與管理系統搭配投影片。課後作業則以紙本與程式解題系統，如高中生程式解題系統、ITSA 程式自學平臺等。同一堂課，師生必須同時使用多種軟體與網站。

- 線上課程：為因應 COVID-19 的疫情，教師常以網路同步直播教學為主，許多基層教師利用 Google Meet 或 Microsoft Teams 等軟體進行線上同步直播教學，然而，這些軟體並不是針對教學而設計的，如作業的批改、實作、課程內容的簡報、回顧等，仍需要使用其他軟體。
- 共同問題：缺乏整合的教學空間，導致學生和教師因教學系統過於複雜而不易使用。

基於以上問題，本計畫旨在通過改進教學方式，幫助教師以最小的成本實現預期的教學效果，從而提高學生的學習體驗和效果。

2. 需求分析與設計

本研究將依循軟體工程流程與方法進行需求擷取與分析、架構與介面設計、系統實作、單元測試、系統功能與效能測試。

(a) 需求分析：

本計畫根據上述之動機與研究問題，擬定多項功能，以符合教師與學生的使用需求，因此我們考慮以下因素：

(1) 自動回饋：

透過自然語言 AI 輔助教學，協助分析學生的程式碼錯誤並給予回饋，以減少教師的工作量，讓教師能更好地關注每一位學生的進度。

(2) 實際操作：

為了增進學生的實際操作機會，本系統將程式實作劃分為課堂實作和課後練習。課堂實作已整合於教學頁面，使學生能在課堂中同步練習並加深記憶。課後練習則透過自然語言 AI 進行自動批改，學生可隨時提交作業並得到批改和建議，教師也可隨時關注學生的問答狀況。

(3) 具互動式：

由於傳統投影片較為靜態的特性，使學生難以深入討論，且由於教師單向輸出內容，使其無法從學生的反饋中，快速檢視學生對於知識的掌握程度。為了改善這個問題，本計畫將重新設計投影片系統，以 JS 語法控制投影片上的各種物件，並加入黑板模式，讓特定學生能編輯投影片的內容，使投影片具有雙向反饋的互動性。

(4) 具整合式：

因市面上的在線教學軟件缺乏整合性，教學投影片、在線教學、程式實作和課後練習分別通過多個軟件和網站實現，導致教學過程中，使用者操作和數據傳輸效率不佳。為了解決這個問題，可以採用整合性的在線教學平台，使教師可以在同一個平台上，上傳投影片、發布直播、建立練習題目，同時使用者可以在同一個頁面上完成教學和練習，不需要過大的操作複雜度，就能實現更流暢的課堂教學環境，並且嵌入式的互動式工具，也能提升學生對於學習的積極度 [15]。

根據以上因素，統整出各項功能：

A. 班級與課程功能

A-1. 教師可新增與查詢自己的課程，並於特定課程中新增章節。

A-2. 教師可於特定章節中查看此章節中，學生的答題統計、問答紀錄。

B. 影音與直播功能

B-1. 教師可於章節中上傳影音。

B-2. 教師可於章節中開啟即時直播，並且能預覽與調整直播設定。

B-3. 學生可於教學頁面中觀看此章節的教學影音。

C. 投影片功能

C-1. 教師可編輯章節的投影片，並於投影片中新增與編輯文字、圖形與互動式元件。

C-2. 教師可開啟黑板模式，讓特定學生能在投影片中繪圖寫字。

C-3. 學生可於教學頁面中觀看此章節的投影片，並與互動性元件互動，如勾選方框、按下按鈕等。

D. 練習題功能

D-1. 教師可於特定章節中新增題目，並且可以在現有題目系統 (如: AtCoder) 直接抓取題目。

D-2. 教師可於特定章節中查看各學生於特定作業的答題情況，內容包含繳交的程式碼、AI 批改的紀錄與分數。

D-3. 教師可針對答題情況，留下教師建議。

D-4. 學生可於特定章節中查看並解決此章節的課後練習題目，學生繳交程式碼後由 AI 自動批改、評分並提出建議。

E. 教學頁面功能

E-1. 學生可於投影片區觀看並操作投影片。

E-2. 學生可於程式區編輯 JS 程式碼，並且能執行與查看結果。

E-3. 教師可查看各學生執行的程式碼與結果。

E-4. 教師可控制黑板模式並使用互動性功能。

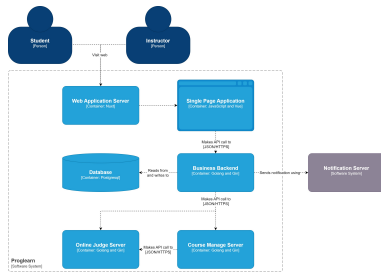
(b) 系統設計

根據需求擷取與分析所述，系統架構如圖1a所示，在 Proglearn 系統內 (如圖1a)，包括前端服務器 (Web Application Server、Single Page Application)、主要後端服務器 (Business Backend)、課程管理服務器 (Course Manage Server)、線上解題服務器 (Online Judge Server)、資料庫 (Database) 等容器。

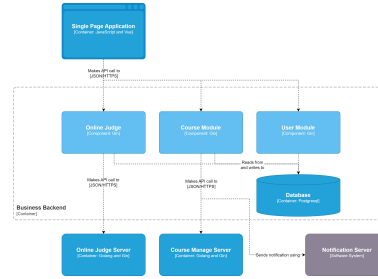
主要後端服務器 (如圖1b) 負責處理與管理使用者資料 (User Module)、課程模組 (Course Module)、題目模組 (Online Judge)。

課程管理服務器 (如圖1c) 包括影音服務器 (Video Server)、投影片模組 (CodeSlides Module)、練習模組 (Exercise Module)。投影片模組將使用 CodeMirror 作為編輯器，並且提供使用 JavaScript 程式碼撰寫投影片。前端呈現如圖2，學生能夠同時看到教師的畫面及投影片。

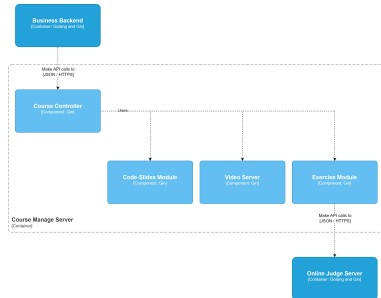
整合式線上教學平台 ProgLearn



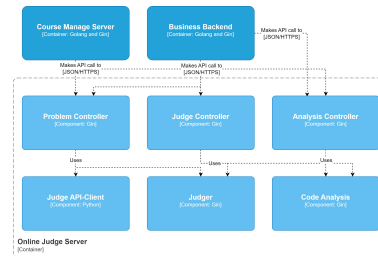
(a) Proglearn 系統架構



(b) 主要後端服務器架構



(c) 課程管理服務器架構



(d) 線上解題服務器架構

圖 1. 系統架構設計 (點擊可看大圖)

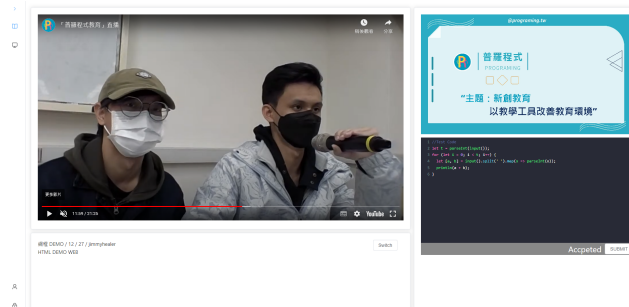


圖 2. 前端教學頁面示意圖 (點擊可看大圖)

線上解題服務器 (如圖1d) 包括線上題目抓取模組 (Judge API-Client)、程式碼批改模組 (Judger)、程式碼分析模組 (Code Analysis)。Judge API-Client 是使用開源專案 api-client [16] 作為線上題目抓取模組，方便教師可以使用現有的題目做修改。Judger 是參考開源專案 go-judge [17]、JudgeServer [18] 開發出沙盒程式碼執行環境及程式碼批改模組。Code Analysis 將採用多標籤辨識模型實現，後續章節將針對此核心技術進行更詳細的說明。

3. 開發技術介紹

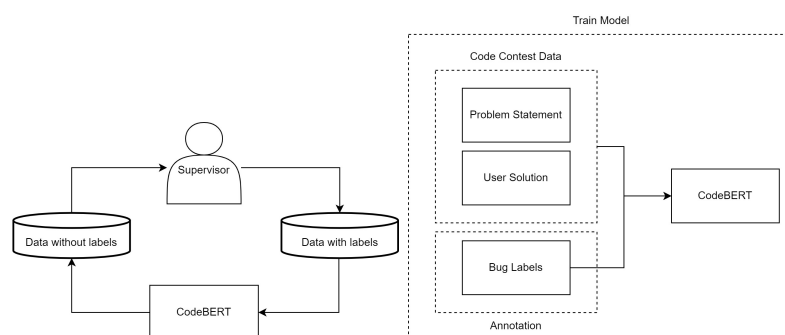
(a) Code Analysis 技術設計

(1) 資料準備

資料來源於 deepmind 開發 AlphaCode 所使用的資料集 code_contests。資料集中包含超過 10,000 個題目和程式碼，包含了 AOJ¹、Codeforces、AtCoder 等。資料集中會包含題目和使用者的程式碼，並且將使用 C++、Python、Java 三個常用競賽程式語言作為訓練資料。首先我們會先將資料集中的題目依照難度進行分類，將題目分為 5 個等級，分別為 A、B、C、D、E。我們計劃整理大約 100 個題目（題目難度分布為 $A : B : C : D : E = 4 : 2 : 2 : 1 : 1$ ），每個題目都會準備約 100 筆程式碼。我們會把這些程式碼分為正確的和錯誤的兩類，比例為 1:9。

(2) 實驗過程

本研究規劃實驗步驟如圖3a所示。首先，我們會將訓練資料集中篩選出 10 題難度為 A 的題目作為初始訓練資料，這些初始訓練資料我們會先標記多個錯誤標籤，錯誤標籤包含：「變數輸入錯誤」(VIE)、「輸出格式錯誤」(OFE)、「演算法時間複雜度過高」(ATH)、「變數未初始化」(VAI)、「變數型態錯誤」(VTE)、「演算法錯誤」(AE)、「題意誤解」(PM) 等。接著，將這些初始資料切割成 8:2 的比例，其中 8 批資料作為訓練資料，2 批資料作為驗證資料。而如圖3b所示，會將題目資訊、程式碼、錯誤標籤作為輸入，並交給 CodeBERT² 訓練。接著訓練完的模型會把剩餘的訓練資料進行預測，而我們會排序信心分數最低的 20 筆資料進行人工審查與標記，並將這些資料加入訓練資料中，並重複上述步驟，直到訓練資料中已審查過的資料都被標記為正確。



(a) 主動學習

(b) 模型訓練

圖 3. AI 訓練實驗步驟 (點擊可看大圖)

(3) 自動反饋使用流程

本研究規劃的自動反饋參考了 Dong 等作者的論文 [14]，使用流程如圖4所示。當學生 (StudentA) 在課堂練習或作業提交程式碼時，會將程式碼交給 Judger 進行批改，若是所有的測試資料都通過驗證，Judger 會回傳一個 Accepted 資訊給結果統計 (Result Statistics)，若沒有通過驗證，Judger 會回傳一個 Rejected 資訊，並且將程式碼交給 Code Analysis 做錯誤分析。接

¹Aizu Online Judge, <https://judge.u-aizu.ac.jp>

²<https://github.com/microsoft/CodeBERT>

著，Code Analysis 分析後將錯誤標籤給結果統計。結果統計會將資訊傳給 StudentA，並且會將資料統整後傳給教師 (Teacher)。前端呈現如圖5所示，教師可以看到某題所有學生的答題情況，掌握學生的學習狀況。

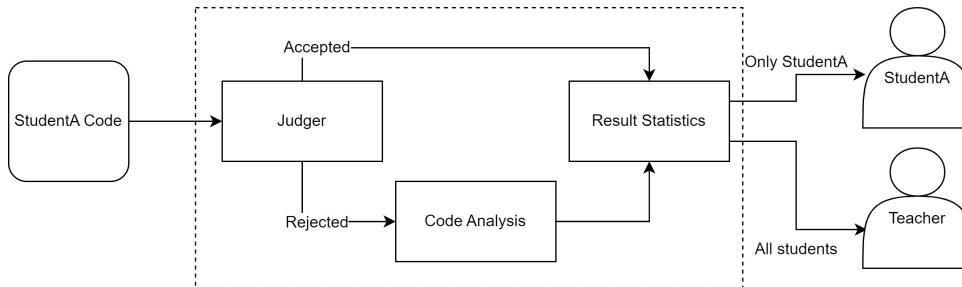


圖 4. 自動反饋使用流程 (點擊可看大圖)

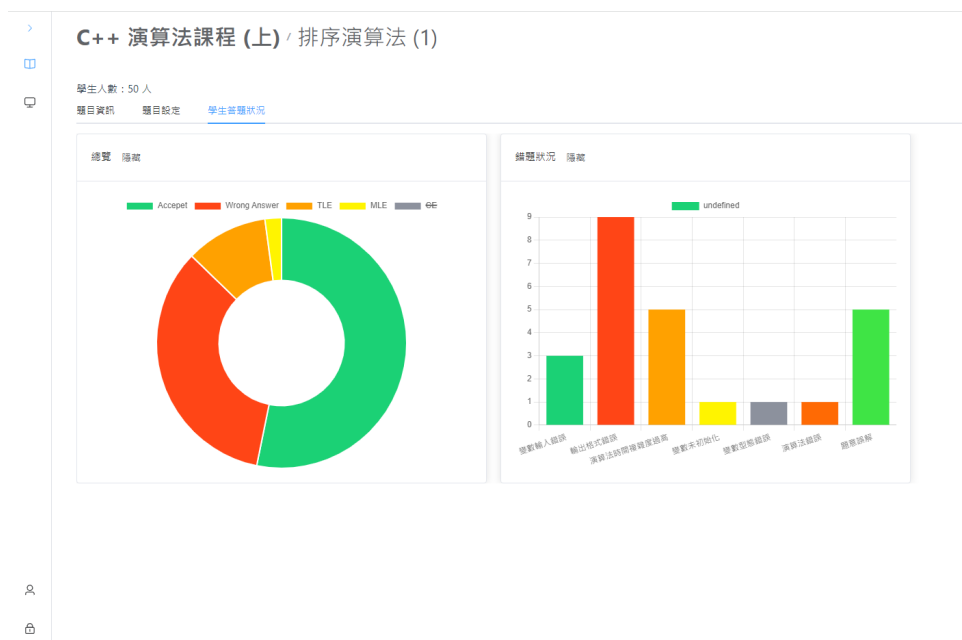


圖 5. 前端自動反饋介面 (點擊可看大圖)

(4) 低延遲直播技術

此研究旨在開發一個低延遲的同步教學平台。這個平台將支持分享電腦螢幕和人像直播功能。市面上常見的視訊串流直播技術：RTMP (Real-Time Messaging Protocol)、HLS (HTTP Live Streaming)、DRM (Digital Rights Management)、WebRTC (Web Real-Time Communication)。RTMP 技術長用於線上遊戲和多人直播。HLS 是使用 HTTP 協議，在不同的網絡環境中提供高質量的實時串流。DRM 技術用於保護數位內容，通常用於付費串流平台中，以防止內容被盜。而 WebRTC 是一個用於即時通信的方案，如解題直播等需要高互動性的情景。

在 Ouya 等作者的論文中 [10]，作者探討了使用 WebRTC 技術來實現遠端教育的平台。WebRTC 是 google 的一個開源項目，用於網絡瀏覽器之間的實時通信。它使用一個 JS API，直接在瀏覽器中實現視訊通話、文件傳輸、P2P 訊息分享和其他功能，而不需要額外的插件或程式。憑藉其現代軟體技術，如資料通道和串流傳輸，WebRTC 提供了一個高效、可靠和可擴展的即時通信解決方案。在我們的 Proglearn 所規劃的教學平台，我們是需要 WebRTC 此技術，解決高互動性的使用情境。

本研究預計使用 WebRTC 結合 RTSP 的概念，來開發出超低延遲直播教學平台。

4. 作品展示

5. 未來與展望

最後我們開發出一個整合式教學系統，將有以下四點特色：

- (a) 自動回饋系統能夠節省教師的時間，教師可以快速了解學生的學習進度和弱點，有更多的時間去創造更好的學習環境和教學資源，解決教師在教學與備課的工作量大。
- (b) 透過同步練習與課後練習並輔以 AI 自動批改，讓教師能夠快速掌握學生的學習狀況，提高了教學效率。
- (c) 互動式簡報模組，可以讓教師與學生之前有更多的互動，讓學生能夠更加主動地參與學習，並且能夠得到及時的回饋和指導。
- (d) 整合式系統可以讓教師在同一個平台上創建和管理課程、學生，並能夠快速分析學生的學習狀況，提高了教學效率。

6. 參考文獻

- [1] 十二年國民基本教育課程綱要。民 112 年 2 月 14 日，取自：<https://www.naer.edu.tw/PageSyllabus?fid=52>。
- [2] 政府資料公開平台（民 111 年 6 月 29 日）。全臺灣各級學校之學生數及畢業生數資料。民 112 年 2 月 14 日，取自：<https://data.gov.tw/dataset/31436>。
- [3] 聯合新聞網（民 110 年 3 月 8 日）。中小學資訊教師荒！多校找自然師兼任遭批不專業。民 112 年 2 月 14 日，取自：<https://udn.com/news/story/6885/5303319>。
- [4] 張瑞賓、李建華。”遠距教學常態化問題之探討與建議。”臺灣教育評論月刊 10.6 (2021): 27-34。
- [5] 岳修平、梁朝雲。”綜整學生，教師與教學情境考量的遠距教學預測模型。”教育資料與圖書館學 52.1 (2015): 33-57+。
- [6] De Giusti, Armando. "Book review: Policy brief: Education during COVID-19 and beyond." Revista Iberoamericana de Tecnología En Educación y Educación En Tecnología 26 (2020): 110-111.
- [7] Tumwesige, Josephine. "COVID-19 Educational disruption and response: Rethinking e-Learning in Uganda." University of Cambridge (2020).

- [8] Santos, Joseline M., and Rowell DR Castro. "Technological Pedagogical content knowledge (TPACK) in action: Application of learning in the classroom by pre-service teachers (PST)." *Social Sciences & Humanities Open* 3.1 (2021): 100110.
- [9] Ratheeswari, K. "Information communication technology in education." *Journal of Applied and Advanced research* 3.1 (2018): 45-47.
- [10] Ouya, Samuel, et al. "WebRTC platform proposition as a support to the educational system of universities in a limited Internet connection context." 2015 5th World Congress on Information and Communication Technologies (WICT). IEEE, 2015.
- [11] Ibrahim, Mohamed, and Osama Al-Shara. "Impact of interactive learning on knowledge retention." *Lecture Notes in Computer Science* 4558 (2007): 347.
- [12] Akram, Huma, et al. "Technology integration in higher education during COVID-19: An assessment of online teaching competencies through technological pedagogical content knowledge model." *Frontiers in psychology* 12 (2021): 736522.
- [13] Krusche, Stephan, and Andreas Seitz. "Artemis: An automatic assessment management system for interactive learning." *Proceedings of the 49th ACM technical symposium on computer science education*. 2018.
- [14] Dong, Yu, Jingyang Hou, and Xuesong Lu. "An intelligent online judge system for programming training." *Database Systems for Advanced Applications: 25th International Conference, DASFAA 2020, Jeju, South Korea, September 24–27, 2020, Proceedings, Part III* 25. Springer International Publishing, 2020.
- [15] Delen, Erhan, Jeffrey Liew, and Victor Willson. "Effects of interactivity and instructional scaffolding on learning: Self-regulation in online video-based environments." *Computers & Education* 78 (2014): 312-320.
- [16] api-client. Available from: <https://github.com/online-judge-tools/api-client>
- [17] go-judge. Available from: <https://github.com/crilyle/go-judge>
- [18] JudgeServer. Available from: <https://github.com/helsonxiao/JudgeServer>