

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) 創作動機與背景

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。

(b) 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工具，讓所有人都能做程式教學。

(c) 研究問題與專業領域

根據教師的教學需求，整理出四個主要問題：

(1) 教師的教學有諸多限制：

程式設計是實作很重要的課程，常見的線下教學方式需要使用廣播與管理系統，因為這些系統無法在授課時即時讓學生練習，導致老師不得暫停授課，給學生練習時間。這種情況可能使教師無法流暢教學，同時學生也難以吸收所學知識。

(2) 教師需要花多時間備課：

教師需要在上課前，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 [3]。

(3) 教師無法即時得知學生的狀況：

由於師生間缺乏互動的方式，讓學生難以保持學習熱忱，同時教師也無法即時注意學生的學習狀況，而影響教學成效 [4]。

(4) 教學工具零散：

在程式教學中，教師需要使用多種不同的工具和平台：準備教學教材的工具，如 PowerPoint、HackMD 等。準備課堂教學的環境，如 Google Meet。以及繳交程式作業的系統，如 ITSA 程式自學平臺、TronClass 等。

由於這些軟體，並非針對教師的教學所設計，導致如批改、實作、設計教材、直播、上課，仍需要透過其他軟體或平台來實現。缺乏整合的教學工具，不但增加了環境的建置難度，也提高教師的教學負擔。

基於以上問題，本系統跨足教育領域，通過改進教師的教學方式，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。

(d) 創新創意

本系統為符合教師的教學需求，設計出以下幾種創新功能：

(1) 課堂直播

- 即時同步的投影片：此直播實際上是一個投影片，可以將老師對投影片的繪畫、換頁等操作同步到學生的畫面中。
- 可隨時切換的投影片：讓學生在直播期間，可以按照自己的學習速度切換到其他簡報，提供學生學習的彈性。

(2) 互動式講義

- 滾動式講義：透過 Markdown 呈現課程講義，讓學生更方便地查看課程講義，也加快教師編寫教材的速度。
- 跟隨模式：課程直播時，在學生的頁面中會以淺黃色提示老師正在教學的內容。
- 嵌入式的程式問題：讓學生能在講義區直接作答，並且能即時得知答案是否正確，教師也可以隨時得到學生的答題狀態。



圖 1. 互動式講義示意圖 (點擊可看大圖)

(3) 學生的回看功能

- 學生在直播中，可以任意切斷到其他投影片，並隨時能切換到老師目前的進度。
- 學生在課後，可以再次觀看過往的上課直播，當學生切換到其他投影片時，時間軸會自動暫停。播放時，可以讓學生回到上次的進度。

(e) 主要功能

該系統提供課堂教學系統和課後作業系統。課堂教學系統提供投影片直播、投影片繪畫、互動式講義、學生的答題統計的功能。課後作業系統提供自動批改程式碼作業的功能。

(f) 實用性與預期貢獻

協助教師提高教學效率，並且提高學生的學習效果。在未來能將此系統提供給如學校、補習班、講座等老師使用。

2. 需求分析

本計畫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。

(a) 功能性需求：

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. 直播投影片功能

B-1. 教師可於特定章節中上傳課程投影片。

B-2. 教師可於教學頁面中直播模式，使學生的投影區畫面與教師相同。

B-3. 教師可於教學頁面中

B-4. 學生可於教學頁面中觀看此章節的投影片。

C. 講義功能

C-1. 教師可編輯章節的講義，並於講義中新增與編輯文字、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. 答題功能

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: At-Coder) 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。

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) 非功能性需求

A. 使用者介面與人為因素

A-1. 使用者：程式設計課程的老師、學生

A-2. 使用者的介面設計：簡單並且易上手、高度的功能整合、即時獲得使用者的反饋與資訊。

A-3. 使用者的引導與教學：直觀的 UI 設計並且對所有老師做系統的使用培訓。

B. 硬體

B-1. 使用者的使用設備：使用設備以電腦為主、手持設備為輔，並針對電腦使用最佳化。

B-2. 使用者的設備限制：所有能瀏覽網頁的設備皆可使用。

C. 效能

C-1. 反應時間：同一課程能讓 100 人同時使用，並在 0.5 秒內回應使用者，以保障使用者的上課體驗。

C-2. 容量限制：限制每位教師在單一章節中，只能上傳 20MB 以下的投影片。

C-3. 課堂限制：每位教師的課程需要經過管理員審核後才能開課，並且每位教師以開設 5 門課為限。

C-4. 課堂人數：同一課程最多只能有 100 位學生。

D. 錯誤處理：

D-1. 系統遇到不正常的負載：針對大量請求的用戶限制封包的流量。

D-2. 系統遇到高負載：使用排隊來限制同時登入人數。

(c) 系統設計

本系統是架設於網路上的 Web 應用系統，以下將針對系統的設計分為系統架構、系統介面說明：

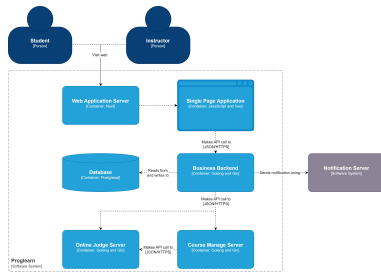
(1) 系統架構

根據需求擷取與分析所述，系統架構如圖2a所示，在 Proglearn 系統內 (如圖2a)，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。

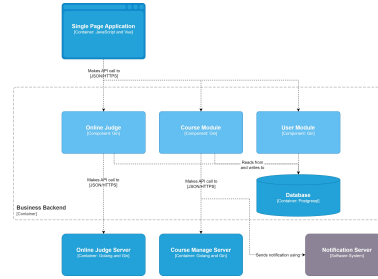
主要後端服務器 (如圖2b) 負責處理與管理使用者資料 (User Module)、課程模組 (Course Module)、題目模組 (Online Judge)。

課程管理服務器 (如圖2c) 包括影音服務器 (Video Server)、投影片模組 (Code-Slides 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3，學生能夠同時看到教師的畫面及投影片。

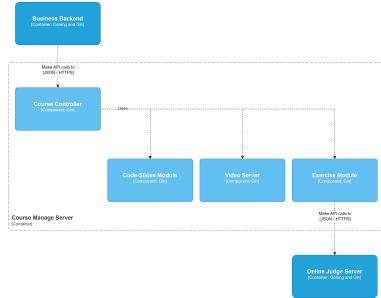
整合式線上教學平台 ProgLearn



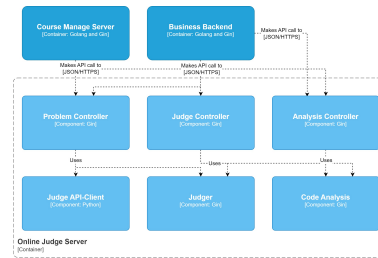
(a) Proglearn 系統架構



(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 2. 系統架構設計 (點擊可看大圖)

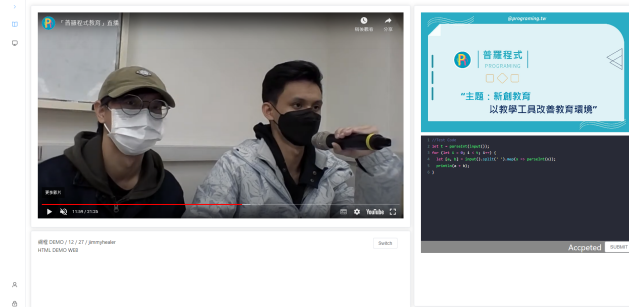


圖 3. 前端教學頁面示意圖 (點擊可看大圖)

線上解題服務器 (如圖2d) 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 [5] 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 [6]、JudgeServer [7] 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。

(2) 系統介面設計

系統的使用頁面主要為課堂教學頁面。並針對不同的使用者，設計兩種使用介面：

- 教師：能夠管理投影區與互動區，並取得學生的答題狀況。

- 學生：能夠觀看投影片、互動區的內容，互動區包含滾拉式的 Markdown 講義，並且嵌入以程式為主的互動性問題，讓學生作答。

3. 開發技術介紹

- (1) 程式語言：TypeScript, Golang
- (2) 前端設計：Vue3, Element Plus, chart.js, Codemirror, Vue Router, Pinia
- (3) 後端設計：Gin, Websocket, WebRTC
- (4) 資料庫：PostgreSQL
- (5) 開發工具：Git, Docker, VSCode, Postman

4. 作品展示

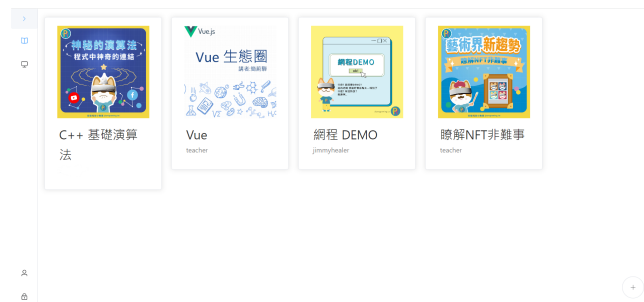


圖 4. 課程清單頁面 (點擊可看大圖)



圖 5. 課程資訊頁面 (點擊可看大圖)

整合式線上教學平台 ProgLearn

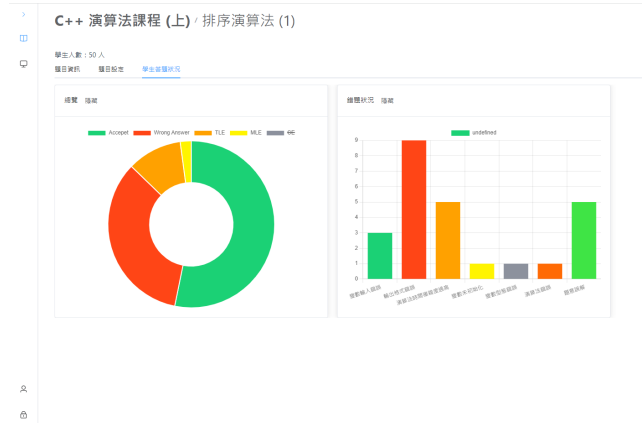
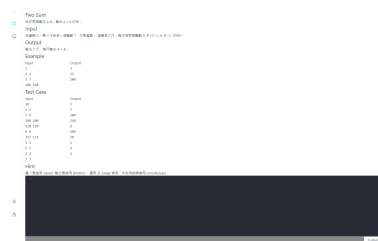
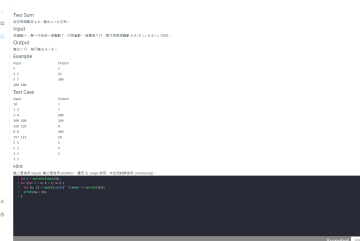


圖 6. 課程作業回饋 (點擊可看大圖)



(a) 題目



(b) ac



(c) wa



(d) re

圖 7. 課後作業系統 (點擊可看大圖)

5. 未來與展望

未來將增加與改善以下四種功能：

- (a) 增加自動回饋系統，讓教師可以快速了解學生的學習進度和弱點，減少教師在教學與備課的工作量。
- (b) 增加智能助教系統，自動批改學生作業並提出建議，讓教師能夠快速掌握學生的學習狀況。
- (c) 增加互動式講義的功能，讓教師與學生之間有更多互動，將抽象的概念視覺化，讓學生更容易理解並加深印象。

- (d) 增加教材的輔助 AI，能夠根據教師的教學內容，提供文字補全與編寫建議，以提高教師編寫教材的效率。

並且以創業與經營為目標，提供教學系統服務給程式教育的教師，

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] 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。
- [4] 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+。
- [5] api-client. Available from: <https://github.com/online-judge-tools/api-client>
- [6] go-judge. Available from: <https://github.com/crilyle/go-judge>
- [7] JudgeServer. Available from: <https://github.com/helsonxiao/JudgeServer>