

參賽隊名：普羅程式

作品名稱：威力導師 PowerTeacher

1. 競賽主題：數位永續科技組

2. 創作主題

(a) 題目

威力導師 PowerTeacher

(b) 實用功能描述

在台灣，資訊科技領域備受關注，且程式設計已成為學校必修課程。儘管每年有數百萬學生修習程式課程，但基層教育仍面臨著許多問題：以往的教學方式需頻繁切換畫面給學生練習、課堂上教師無法即時得知學生狀況。所以我們建立一個名為 PowerTeacher 的網頁應用，專注於教師導向的程式教學工具，整合直播、實作、測驗、互動與 AI 輔助，讓所有人都能進行優質的程式教學。

PowerTeacher 是專為教師設計的教學工具，採網頁應用形式，主要提供三種頁面：

(1) 學生課堂頁面：學生的學生課堂頁面，有直播區、互動區、功能區（見圖 x）。

- 直播區：用於顯示章節投影片，會在課中展示與老師相同的投影片畫面，並同步老師的滑鼠軌跡、繪畫等。
- 互動區：用於顯示滾動式的講義，講義可以放文字、圖片、課堂習題，並且在課中，具有引導功能，會根據老師目前的上課投影片，用黃色框線在講義中顯示其對應的位置。
- 功能區：用於控制直播區的內容，在課中能夠切換投影片、一鍵回到老師的直播投影片等。在課後能夠拖動時間軸，回放過去的上課直播。

(2) 教師課堂頁面：教師的課堂頁面，與學生的課堂頁面相同，同樣有直播區、互動區、功能區，但功能有部分差異（見圖 x）。

- 直播區：用於顯示章節投影片，在課中會將畫面同步到學生的直播區中。
- 互動區：用於顯示滾動式講義，並能夠預覽課堂習題的作答統計。
- 功能區：用於控制直播功能，能夠開啟與關閉直播，在課中可以切換投影片、切換成畫筆功能等。

(3) 教師講義編輯頁面：教師的講義編輯頁面，用於編輯互動區講義的內容，我們將整個講義分為各種不同類型的小區塊，透過將不同類型的區塊做拼接，以完成整個講義。這個構想是參考影片剪輯軟體，能夠在時間軸上，將各個影音片段組合成一部影片。分為編輯區與時間軸（見圖 x）：

- 編輯區：在右半部點擊不同類型的講義區塊，如文字、選擇題、程式題，就能夠在左半部編輯其中的內容。
- 腳本區：下半部的時間軸，用於將不同類型的講義小區塊做不同順序的拼接，以完成整個講義。時間軸的單位是投影片的頁數，讓投影片能對應到不同的講義區塊，在課中就能根據投影片的頁數在講義上做引導與提示。

(c) 作品與市場相關產品差異

經過與大學與高中程式教師的實地訪談，我們整理出以下三個時間段的教學流程：

(1) 課前：準備教材

在教學前，教師會準備課堂所需的講義與投影片。市場上的講義編輯功能通常是基於文件編輯器或投影片的形式，例如 Microsoft Word、Power Point。然而這些工具的功能較為單一，沒辦法嵌入程式執行區、互動習題等與教學相關的功能。我們的平台提供滾動式講義，搭配程式執行、引導、互動習題等功能，使講義內容與課堂做連結。並使用教師講義編輯頁面，使講義更為直觀、易於操作和修改。

(2) 課中：互動教學

- 直播功能：實現直播教學的方式，能大致分為硬體與軟體，硬體上常見的有廣播與管理系統，能夠強制控制學生的畫面。軟體上則有 Zoom、Google Meet 等以視訊為主的會議平台或專為學校開發的遠端控制系統，透過網路分享教師的語音與畫面。這些工具分別有幾項問題：前者是強制控制學生電腦，無法讓學生在課堂中與老師同步實作，也無法用電腦查詢資料、觀看講義等。後者是直播的影音可能有延遲，會導致老師的教學與控制不流暢。我們的特色是讓學生能夠在課堂中，操控投影片回顧上課內容，還能同時觀看補充講義、實作程式碼、回答習題等，讓學生就算在課堂中也能回顧與實作，並以更低延遲的直播投影片取代影像直播，使學習更為流暢。
- 引導功能：如何讓學生在課堂中更有參與感並且理解教學內容。市場上的一般教學軟體或平台通常缺乏對於學生學習的引導功能。我們的平台嘗試解決這個問題，透過互動區的黃色框線，在講義中顯示對應的位置，讓學生能夠清楚知道老師目前講解的內容與講義之間的對應。這樣的引導功能可以讓學生更容易理解並隨著教學進度進行，同時也能在回顧時更加方便。

(3) 課後：課程回顧

在市場上，許多教學平台提供課後回放功能，讓學生能夠在課程結束後回顧老師的教學內容。我們的平台也提供了這樣的功能，讓學生能夠在課後拖動時間軸，回放過去的上課直播，以便進一步學習和復習。不過，我們的特色在於，課後回放不僅僅限於觀看直播畫面，還能夠觀看補充講義、實作程式碼等，並搭配引導功能，讓學生在回顧時更為全面與深入。

綜合以上幾點，我們整理了老師上課時可能會用到的工具並做比較：

本系統在直播延遲、教學方式、課後回顧、線上練習以及教學功能整合方面表現出色，並減少對學生電腦的控制，提供了較佳的教學彈性和互動。

功能	本系統	Google Meet	遠端控制系統	CodingBar	廣播與管理系統
直播延遲	低	高	高	高	低
教學方式 <sup>1</sup>	線上與實體皆可	線上	實體	線上	實體
電腦控制			遠端控制 <sup>2</sup>		完全控制
課後回顧	✓			✓	
線上練習 <sup>3</sup>	✓			✓	
教學功能整合 <sup>4</sup>	✓			✓	

Table 1: 老師上課時可能會用到的工具比較

### 3. 創意構想

- (a) 理論基礎
- (b) 設計創新說明
- (c) 特殊功能描述

<sup>1</sup>教學方式：教學方式分為線上與實體教學，線上教學指的是在網路平台上進行的教學活動，通常包括遠距視訊教學、線上課程和互動學習模式。實體教學指的是在實際的物理教室或學習環境中進行的教學活動，教師與學生面對面進行互動和知識傳遞。

<sup>2</sup>遠端控制：遠端操作是指教師可以強制操作學生的電腦，或者同時與學生在同一台電腦上進行操作。

<sup>3</sup>線上練習：線上練習是指讓學生能夠在線上環境中進行練習、測試和應用所學的知識或技能。這種方式可以包括線上測驗、程式撰寫與評測等。

<sup>4</sup>教學功能整合：教學功能整合是指將不同的教學元素、工具和方法結合在一起，可以是教學資源、互動工具、直播平台等。

# 1 系統架構

## 1.1 架構說明

### 1.1.1 前端 (Vue.js)

使用者界面將採用 Vue.js 框架來開發，前端主要包括以下幾個部分：

- 直播界面：提供課程直播及實時跟隨課程進度的功能。
- 互動式講義：教師可編輯，學生可同步學習並完成練習題。
- 作業區 (Online Judge)：學生可做題並獲得即時反饋，教師可查看結果及統計。

### 1.1.2 後端 (Gin)

後端部分我們將採用 Golang 的 Gin 框架來開發，後端主要負責以下功能：

- API 提供：為前端提供 RESTful API，如獲取課程列表、課程詳情、新增課程等。
- 課程管理：儲存和管理課程及章節資訊，並回應前端的請求。
- 互動講義管理：處理講義相關的請求，如顯示講義、新增練習題等。
- 題目管理：儲存和管理作業及互動式講義練習題資訊，並與 judge 系統互動。

### 1.1.3 資料庫 (PostgreSQL)

我們選擇使用 PostgreSQL 作為我們的主要資料庫，資料庫將儲存以下資訊：

- 課程資料：包含課程的基本資訊如課程名稱、課程描述、課程章節等。
- 章節資料：包含每個章節的基本資訊如章節名稱、章節內容、互動題目等。
- 講義資料：包含講義的基本資訊如講義名稱、內容、相關章節、互動性元素等。
- 題目資料：包含練習題和互動題的基本資訊，以及用戶的解答狀態和成績等。

### 1.1.4 直播系統 (WebRTC + WebSocket)

在我們的平台中，實時課程直播服務將主要聚焦在傳輸老師的聲音以及與學生的互動功能，主要使用以下兩種技術：

- WebRTC：使用 WebRTC 技術傳輸老師的聲音，實現網頁間的實時通訊。
- WebSocket：使用 WebSocket 技術傳輸老師的滑鼠動作和教學講義的同步滾動，提供全雙工的通訊通道，實現服務器和客戶端的即時通訊。

### 1.1.5 judge 系統 (Judger0)

judge 系統使用 Judger0 作為基礎，並提供一個安全的沙盒環境來執行和測試程式碼。它能夠與後端的題目管理互動，使得可以讓學生於前端提交程式碼來解決特定的練習題，教師可以使用此功能來評估學生的程式設計能力和理解程度。

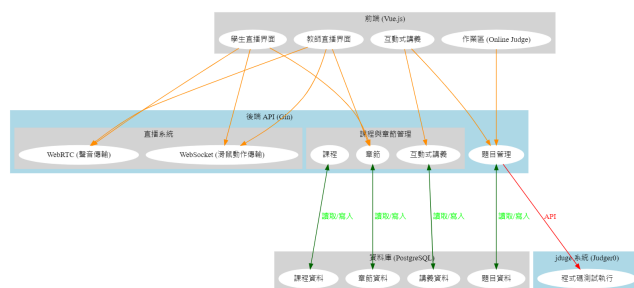


圖 1. 系統架構圖

這樣的架構設計是為了使 PowerTeacher 更加模組化和可擴展性，同時能確保每個部分的功能得到最佳實現。例如，前端和後端的分離使得前後端的開發可以並行進行，大大提高了開發效率。選擇 PostgreSQL 是考慮到其高度的可擴展性和穩定性。直播系統的設計則是為了實現更好的即時教學體驗。這套架構能夠滿足 PowerTeacher 的各種需求，並為未來的擴展打下良好的基礎。

## 1.2 「人機介面設計」(UI) 與「使用者體驗」(UX) 設計

以下使用 Jakob Nielsen 使用者體驗設計優化原則說明：

### 1.2.1 Visibility of system status

- 所有課程和章節：使用者在開啟或編輯課程或章節時，系統顯示載入或操作進度。
- 投影片上傳：系統在教師上傳教材時顯示上傳進度，並在完成後確認。
- 教學頁面：系統在直播進行時顯示播放進度，並在中斷時立即通知使用者。

### 1.2.2 Match between system and the real world

- 所有課程和章節：課程和章節的結構仿照了真實的教學環境。
- 互動式講義：模仿真實的課堂環境，包括教師講解和學生練習。

### 1.2.3 User control and freedom

- 所有課程和章節：使用者可以自由瀏覽所有參加的課程和已學習的章節，不受限於一定的順序或路徑。
- 教學頁面：在觀看回放錄影，使用者都可以隨時暫停、倒退或快進；可自由查看直播、章節概述和互動式講義。

### 1.2.4 Consistency and standards

使用 Element Plus UI library 實現了這一原則。

### 1.2.5 Error prevention

- 課程與章節管理：系統檢查教師輸入的資訊是否完整和合規，例如確認課程名稱是否已存在，或章節內容是否為空。
- 互動式講義：在使用者提交答案前，系統檢查答案格式並提醒使用者確認，避免意外提交。
- 投影片上傳：系統驗證上傳檔案的格式和大小，並提供取消上傳的選項，避免非預期的上傳。

### 1.2.6 Recognition rather than recall

- 課程與章節管理：使用者可以方便地查看已註冊的課程和已學習的章節，並透過搜尋和篩選功能尋找特定課程或章節，降低記憶負荷。
- 互動式講義：透過高亮協助學生回憶上次的學習位置，降低尋找學習進度的時間。
- 投影片上傳：提供清晰的列表顯示已上傳的投影片或教材，讓教師迅速找到所需的資源，無需記憶與搜尋。

### 1.2.7 Flexibility and efficiency of use

- 課程與章節管理：快速搜尋和過濾功能，使查找課程更有效率；教師的自由新增和編輯功能，增強了管理課程和章節的靈活性。
- 互動式講義：自動儲存學習進度，讓學生能快速回到學習進程；教師可根據課程內容自由編輯講義，提供了教學靈活性。
- 投影片上傳：一鍵上傳和管理投影片或教材的功能；支持多種文件格式的上傳，確保了使用的靈活性。
- 教學頁面：教師可以自由安排直播時間和內容，提供了教學的靈活性。

### 1.2.8 Aesthetic and minimalist design

使用 Element Plus UI library 實現了這一原則。

### 1.2.9 Help users recognize, diagnose, and recover from errors

- 所有課程和所有章節：如果某課程或章節不存在，提供明確錯誤訊息（例如「該課程不存在」或「該章節不存在」），並提供回到上一頁或主頁的選項，讓使用者快速恢復。
- 互動式講義：提交答案不正確時，提供明確反饋（例如「答案錯誤」或「請再試一次」），並提供解答和解釋，幫助使用者理解錯誤原因及正確答案。
- 投影片上傳：如上傳文件格式不支持或文件大小超過限制，提供明確錯誤訊息（例如「不支持的文件格式」或「文件大小超過限制」），並指導如何選擇適合的文件。
- 教學頁面：直播中斷或其他問題發生時，提供明確錯誤訊息（例如「直播已中斷，請稍後再試」），並提供可能的解決方法（例如重新加載頁面或檢查網路連接）。

### 1.2.10 Help and documentation

- 課程與章節管理：課程和章節資訊清晰展示，讓使用者快速了解課程內容和章節進度。提供搜尋和篩選功能，便於找到感興趣的課程或特定章節。
- 互動式講義：提供目錄和標籤功能，讓使用者快速找到查看或復習的內容。確保互動題目的正確答案和解釋清晰展示，便於學生自我檢查和學習。
- 幫助文件：我們會提供幫助文件，紀錄如何使用系統並包含 Q&A。

#### 1. 計劃管理

工作階段	工作日數	工作內容
1		

Table 2: 計劃管理

	1	2	3	4	5	6	7	8
工作階段 1								
工作階段 2								
工作階段 3								
工作階段 4								

圖 2. 甘特圖

#### 2. 修改舊作參賽說明

本專案開發之作品未使用團隊成員曾獲競賽獎勵之作品。

#### 3. 軟體清單

- 作業系統環境：Windows、Linux
- 主要開發程式語言：Javascript、Golang
- 專案支援語言：中文
- 開發環境：
  - Visual Studio Code
  - Node.js
  - Node Package Manager
  - Vue 3 Frontend Framework
  - Gin Backend Framework
  - Git, Github

(e) 專案成果預定授權條款：

本專案開發產品授權條款使用 CC BY-NC 4.0 宣告。

4. 權力分配

依著作權法第 40 條之規定，由參賽學生與指導教授均等共有。