

2026 台灣 AI 教育年會教學應用投稿文件

主題名稱：不只 Vibe Coding：教師以 AI 協作打造專案式學習評分系統的實踐報告

設計者：楊可倫（新北市立林口高中）

設計理念

一、AI 應用設計的起點與問題陳述（必填）

設計起點/動機：

這是一個教師在 LLM 輔助下開發用於教學的專案管理 / 評分系統的故事。

在強調自主學習與探究式學習的今日，社會科——尤其是歷史科——的課堂產出已經不再是一份報告或一疊考卷。規劃專案、執行專案、發表專案，正在成為學生的必備能力；而多元且動態的成果，也帶來了一個核心問題：專案怎麼管？

老師的問題說穿了是流量問題：專案階段切割得越精細，批改流量就越龐雜；引入團體協作後，「版本地獄」（version hell）隨之而來，加上互評更加複雜。

筆者非資訊科系出身，但習慣在工具不夠用時自己動手做，從高中時代幫學校架設官網開始，一直到生成式 AI 誕生後，筆者把出題的工作流逐漸轉換到去 Kaggle 抓取歷史數據集，並讓 LLM 生成 d3.js 的統計圖表用於素養題命題。2025 年 7 月，筆者開始和 Claude Opus 4 逐段審閱設計規格、反覆討論，確認可行後才讓 Claude 開始實作，故事也從此開始。

欲解決之核心問題：

教師端

批改負擔重：專案切割成 N 個階段後，批改量會放大成 $N \times M \times Y \times C$ （階段數 \times 指標數 \times 學生數 \times 互評數），開放式成果更難量化。

流程管理繁複：多階段繳交、版本追蹤、組內共識確認，現有工具（Google Classroom 等）無法統一處理。

學生端

互評流於形式：缺乏動力深度閱讀彼此成果，打星評分無法產生真正的學習回饋。

評分黑箱： 互評過程不透明，學生往往抱怨自己的意見被忽視。

現狀分析（選填）：

受限於個資法與教育網路環境，系統採用 serverless 架構，不觸碰學籍個資。現有課堂作業管理軟體（Canvas、Google Classroom）的小組功能不足以應付多階段專案所需：版本管理無硬性規定，教師評分與學生互評缺乏全自動的計算機制，設計不周的互評很容易淪為搭便車的溫床，簡言之，有方法，但是沒有系統性的解決方法。

二、設計中的創新與獨特想法（必填）

核心創新點：

評分民主化：從系統設計的最根本出發

一切的規格、演算法和程式碼都從這一個想法出發：如何讓專案的評分機制更透明、更民主化。

傳統批改的責任都在老師身上，但專案式學習的成果是眾人一起產出的，評分也應該經過眾人參與、得到眾人同意、並且具備可回溯性。這三件事，才是所有設計決策的出發點。

有時我們可以給作業一組規準（rubric），教師按指標打分即可。但專案型學習評的不只是「這份作業好不好」，而是「這個成果比其他人好還是差」——評圖、設計評審、提案競賽，都是比較性的，百分制的規準無法捕捉這種相對優劣。而且成果是分階段產出的，沒有參與某個階段的學生，就不應該拿到該階段的積分——參與本身是計分指標的一部分。

強制排名比積分投資更直接。只要在每個階段設立獎勵點數，排名就可以按比例轉化成點數，搭配組內的分工表，就能按照參與度將點數分配給每一位同學。

用規格書主導 Vibe Coding

確定採用排名取代評分後，下一個問題是：要用哪種排名算法？筆者從比較政治的背景出發，經過兩輪和 Claude 的討論後，確立採用 Borda Count 作為核心投票邏輯——它不會發生多輪選舉，第一名就是所有人共識中拿到最多高分票的人。筆者請 Claude 寫了幾次投票模擬程式驗證可行性，Borda Count 也可以輕易做出視覺化圖表，讓使用者即時看到投票結果如何影響點數分配。

確定投票模組後，再和 AI 一起規劃權限模型與資料庫結構：組員、組長、教師、未分組人員、旁觀者，共五層權限，並涉及通知觸發機制與權限邊界。

Google Apps Script 和 Cloudflare 雙平台支援

系統原本建置於 GAS，但 GAS 算力不足，以 Google Sheet 取代資料庫也會導致密碼安全性下降。筆者讓 Claude 對整個已完成的 GAS 系統進行重構，轉移至 Cloudflare，取得了 SQLite 資料庫與 TypeScript 支援，同時保留原有前端框架。GAS 版本作為備援，讓無伺服器需求的教師也能部署。

公開帳本：用資料視覺化呈現每一步操作過程，讓操作更透明

本系統在設計之初就非常在乎可溯源性與評分透明化，具體體現在六個層次的資料視覺化設計：

專案進度儀表板：所有人都能即時看到每一組的繳交狀態與階段進度，成果與評語公開可查，分數從何而來一目了然。

參與度視覺化：發布成果時，組長需填寫每位組員的參與程度，系統以圖表呈現占比，並即時預覽勾選結果如何影響每位成員在該階段可獲得的點數。

副署與撤回機制：成果發布後，其他組員必須在系統上「副署」確認；若有異議可撤回，系統完整記錄每一次操作，搭便車問題有制度性的制衡。

Diff 版本比對：每次修改繳交內容，系統以 diff 格式標示與上一版的文字差異，讓版本演進過程清晰可查。

階段互評分佈圖：每個階段結算後，所有組別的互評分佈以圖表公開呈現，成績登記不再是老師的私有文件，而是全班共享的公開帳本。

即時天梯圖：學生可隨時查看自己與第一名、最後一名的相對距離，排名變化透明即時，資訊越公開，爭議越少。

AI 輔助批閱：Free-MAD 多智能體辯論排名

系統轉移至 Cloudflare 後，支援 queue 等長時間非同步運算，筆

者因此建置了對多個 AI 的接口。教師批改成果時，可呼叫 AI 輔助批改系統，採用 Free-MAD（Consensus-Free Multi-Agent Debate）架構，多個 AI 對學生成果展開兩輪辯論：

第一輪：各 AI 獨立排名，不互相參考

第二輪：每個 AI 看到其他人的排名後，被明確要求「只有確認自己有明確錯誤才改變立場，不要從眾」

聚合：以加權積分計算（堅持原立場加分、被改變扣分、被他人採納加分），辯論歷程完整記錄可追溯

通用架構，跨課程零成本部署

階段數與繳交內容完全自訂，目前已驗證兩種課程場景，無需重新建置：

高中歷史課 AI 聊天機器人專案（114-1、114-2）：

Obsidian 筆記 → Dify.ai RAG 機器人 → 機器人互評，3 階段

高中遊戲設計課 GDD 專案（114-2）：

設計提案 → 遊戲企劃書，2 階段

設計差異化：

AI 時代最簡單的評量方式，是把學生作業直接上傳給 AI 批改。但這套系統選擇了不同的路——因為 AI 批改的品質，根本上取決於學生作業的品質；而學生作業的品質，取決於流程設計。

這套系統的核心主張是：好的流程才能產出好的資料，好的資料才能讓 AI 真正發揮作用。多階段繳交、組內共識確認、同儕互評的強制參與——這些不只是管理機制，而是在確保每一份進入 AI 批閱的成果，都經過人的思考與篩選。垃圾進，垃圾出；系統的工作，是讓進去的不是垃圾。

市面上評量工具的 AI 功能是後製外掛；這套系統的 AI 參與了每個環節——設計討論、程式開發、批閱建議。更重要的是，它示範了一個可複製的模型：非工程背景的教師，只要能清楚表達需求、與 AI 反覆驗證邏輯，就能打造出生產級的教育工具。

三、具體成效（質化或量化的說明）（必填）

量化成效 (Quantitative Results) :

系統目前已橫跨兩門課程、七個班次、兩個學期：

歷史課 (114-1、114-2) : 212、208、215 班，各班 4–6 組，4 個評分階段，全部結算完畢

遊戲設計課 (114-2) : 1 個跨班專案 (「歷史的遊戲設計」)，4 個評分階段，進行中；另有 215、208、212 三班的學期報告同步進行中

互評參與數據 (以 208 班為例)

提案→筆記→機器人→製作心得，四個階段評論數分別為：1 篇、10 篇。筆記階段評論從 1 篇大幅增加至 10 篇，反映學生在「機器人對抗」前夕主動投入互評的行為轉變——對方筆記品質直接影響機器人對戰表現，評論有了實質誘因。

資安部分 (採用 Cloudflare 的優點)

本系統原本是架設在筆者家裡的電腦，但要維持隨時上線，筆者得自己注意許多資安問題，也得維護 Linux 系統，因此在轉移到 cloudflare 之後，系統穩定度大幅提高，資訊安全方面，本系統導入了國際公認的 OWASP API Security Top 10 (2023) 安全標準，建置了一套自動化的安全檢測機制。我們針對系統後端 API 撰寫了共 292 項安全測試案例，檢測內容包括：身分驗證是否可被繞過、使用者能否越權存取他人資料、系統是否能抵擋 SQL 注入及跨站腳本攻擊 (XSS) 等常見攻擊手法、以及 WebSocket 即時連線的安全性等。確保系統在每次更新後仍維持安全防護水準。

質化成效 (Qualitative Results) :

學生行為改變

歷史課「機器人對抗」階段強迫學生實際使用他人 RAG 機器人

提問，Obsidian 筆記品質自發提升，積分與評語透明化後，學生主動對照高分組的評語修改自己的成果

教師工作流程改善

批改模式從「逐件人工評分」轉為「審閱 AI 辯論結論後做最終判斷」，用版本控制來管理階段成果、組內共識投票、階段推進全部自動化，不再靠 email 或 LINE 追蹤

學生回饋

學期結算後，筆者以問卷蒐集學生對系統的使用回饋。以下三則回應，各自代表不同層次的使用者反應：

「雖然做了很多機制，但是無法防止搭便車的人（人性啊...）」

「應該要設計『同名次』的機制，有些組表現差不多實在很難排名」

「系統應該要提示我現在該做什麼，我常常忘記去確認我們這組的報告；手機版的評分系統總是看不到通知，只能看到一半的頁面」

這些回饋的價值不只在於改進方向——能指出制度邊界、提出結構性建議的學生，是真的理解了這套機制在做什麼。評分不再是黑箱，學生開始以使用者的角度思考系統本身。

四、設計應用時若有參考或使用到下列資源，請依實際狀況填寫

- 主要參考文獻/論文/演講：
 - 投票（計分）機制設計：
 - ◆ NYC Board of Elections (2021). NYC Mayor Race: Test Ballots Wreak Havoc, Lead to 135K Vote 'Discrepancy' in Ranked-Choice Results（IRV 計票脆弱性案例）
 - ◆ NPR (2021). The 'Human Error' That's Snarling The New York City Mayor's Race（RCV 逐輪淘汰機制的系統風險）
 - OWASP 設計
<https://ithelp.ithome.com.tw/articles/10396575>
<https://ithelp.ithome.com.tw/articles/10381286>
 - Cloudflare Workers 的相關分析
<https://ithelp.ithome.com.tw/articles/10380654>
<https://ithelp.ithome.com.tw/articles/10388702>
 - 使用之數據集：無（這是專案管理系統）
 - 開源工具/框架：
 - ◆ 前端框架：[Vue.js](#)+D3.js
 - ◆ 後端框架：Hono Framework
 - ◆ 資料庫：SQLite（Cloudflare 版本又稱為 D1）
 - 其他資源：
Claude 聊出完整 plan 的對話
claude.ai/chat/c69abbd0-dc95-49b4-b9ba-c2c5e5a14e0f
claude.ai/chat/590c2d27-7ac5-475a-beb1-73968e8d1ac6
claude.ai/chat/a3d35ea8-7988-4a85-b524-f741fb6a18c7

五、「附件或相關照片說明」請視需要提供佐證資料

本系統網址

<https://scoring.kelunyang.online/>

完整的學生回饋

時間戳記	電子郵件地址	評分系統建議
1/24/2026 10:56:48	wade98***@gmail.com	雖然做了很多機制，但是無法防止搭便車的人（人性啊...）
1/24/2026 15:38:29	c36***145@gmail.com	我常常忘記去確認我們這組的報告,他的分數是均分的，我們這組人太多了一人只能分一點點 😊
1/24/2026 16:03:18	tho***6247@gmail.com	應該要設計"同名次"的機制（有些組表現差不多實在很難排名）
1/24/2026 16:51:19	nov***031@gmail.com	其實沒什麼建議，這樣覺得就很好ㄌ
1/24/2026 17:05:03	a096869***7@gmail.com	應該要設計"同名次"的機制（有些組表現差不多實在很難排名）
1/24/2026 18:26:52	cat***ne.zyh0410@gmail.com	我覺得系統應該要提示我現在該做什麼（你可以把如何通知的建議寫在其他裡），我常常忘記去確認我們這組的報告,手機版的評分系統總是看不到通知那個通知，只能看到一半的頁面
1/24/2026 20:46:49	0612***yu@gmail.com	我覺得系統應該要提示我現在該做什麼（你可以把如何通知的建議寫在其他裡），應該要設計"同名次"的機制（有些組表現差不多實在很難排名）

本專案的 Github repo

[目前 repo] <https://github.com/kelunyang/scoringSystem-gas>

[舊專案] <https://github.com/kelunyang/scoringSystem-Server>

[舊專案] <https://github.com/kelunyang/scoringSystem-UI>

系統操作畫面

登入 / 兩階段+captcha 認證

× 歡迎使用 歷史課專用評分系統

登入 註冊 忘記密碼

驗證密碼 兩階段驗證

驗證密碼 確認email和密碼 兩階段驗證 輸入驗證碼

驗證碼已發送到 kelunyang@outlook.com
請查收您的電子郵件並輸入12位驗證碼 (連字號可省略)

驗證碼

↓ 2 3 4 - 5 6 7 8 - 9 10 11 12

數字為輸入順序提示，請依序輸入驗證碼

確認驗證碼

重新發送驗證碼 (48s)

成功! CLOUDFLARE

專案主控台

首頁 > 專案

重新整理

楊可倫
52
01:59

專案

錢包

系統管理

215 114-2 學期報告 ● 提案 剩餘6天14小時

完整學期報告規劃，請參考 [這份Google文件](#)

請注意如果你們選的是不在片單，或者是在特別說明區的影片，一定要先跟老師說

開始 > **提案** > 筆記

進入專案
專案錢包
事件日誌

208 114-2 學期報告 ● 提案 剩餘6天14小時

完整學期報告規劃，請參考 [這份Google文件](#)

請注意如果你們選的是不在片單，或者是在特別說明區的影片，一定要先跟老師說

開始 > **提案** > 筆記

進入專案
專案錢包
事件日誌

114-2 歷史的遊戲設計 ● 尚未開始

請參考本課程的大綱 [了解每一階段該交什麼成果](#)

提案 > 遊戲企劃書

進入專案
專案錢包
事件日誌

專案階段主控台

首頁 > 專案

208 114-1學期報告
重新整理
專案事件檢視

楊可倫
52
01:58

專案

錢包

系統管理

檢視時間為 2026/01/14 ± 4 天

左右拖曳可以看到完整的專案階段, 滑鼠滾動改變可視範圍

收合階段時間軸

已完成

查看報告
查看評論
張貼評論
查看獎金分配

提案

本階段應該繳交以下兩個項目 (都要寫在階段... [顯示完整描述](#))

階段報告獎金	階段評論獎金	已繳交成果	已發布評論
400	200	4	1

分組543... 2025/12/28 21:13:34

最後結算名次: **4** 異組獲得點數: **40**

報告參與者:

查看報告

分組9d5... 2025/12/28 17:42:45

最後結算名次: **3** 異組獲得點數: **80**

報告參與者:

查看報告

208C組 2025/12/27 21:18:51

最後結算名次: 異組獲得點數: 報告參與者:

13

Free-MAD 模式

專案詳情 > AI 輔助建議

AI 模型選擇

deepseek-reasoner (deepseek-reasoner) × 直接排名 BT 配對比較

每項目比較次數：3

預計 9 次配對比較，約需 36 秒

自定義提示 (選填，最多 100 字)

這是一個dify.ai的聊天機器人設計專案，目前是第一階段，要繳交選題和拿來做RAG的素材

45 / 100

此提示會附加到 AI 評分標準後

將對 6 個成果進行 AI 排名分析

查詢歷史 共 2 筆

deepseek-reasoner BT 模式 12/30 下午05:31 6 項 12134 tokens kelunyang@outlook.com

基於 9 次配對比較的結果。[979348] 贏了 1 場；[1d84e6] 贏了 2 場；[b62512] 贏了 2 場...

Cloudflare 後台

Requests **539** ↗ 8.67%

Errors **0**

CPU Time **6.48 ms** ↗ 150.35%

Bindings

- scoring-system
 - Queue AI_RANKING_Q...
 - KV namespace CONFIG
 - D1 database DB
 - Queue EMAIL_QUEUE
 - KV namespace KV
 - Queue LOGIN_EVENTS
 - Durable Object NOTIFICATION...
 - Queue NOTIFICATION...

Request Distribution & Placement

PLACEMENT: DEFAULT