

## 競賽切結書

報名團隊保證已確實瞭解「Tech New Stars 科技新秀大賽」之規則，並同意遵守下列各項規定：

- 一、 報名團隊擔保各項報名資料正確無誤，以及競賽內容係報名者之原創理念。
- 二、 參賽作品不可有違反善良風俗、侵害他人之智慧財產權等議題，若有侵害他人智慧財產之違法情事，主辦/協辦/執行單位有權取消其報名資格或得獎資格，報名者並願意負擔一切法律責任。
- 三、 報名團隊如有違反活動規則或其他相關法令規定之情形，主辦/協辦/執行單位有權取消其參賽資格或得獎資格。
- 四、 參賽隊伍應保證所提供之所有資料為真實及正確，簽署應為本人親筆簽名，且作品內無不雅或不當內容。
- 五、 評審進行中如果發生任何技術或非技術之問題，或因其他任何主辦/協辦/執行單位無法掌控之問題，導致影響評審之正常進行，主辦/協辦/執行單位得結束、延後、調整報名者之權利。
- 六、 報名團隊充分瞭解主辦/協辦/執行單位對於逾期、遺失、不完整、寄送錯誤、不合格、無法閱讀之報名作品，不負任何責任。
- 七、 報名團隊同意放棄因本活動而對主辦/協辦/承辦單位及其他贊助支援廠商之任何法律追訴權。
- 八、 報名團隊同意主辦/協辦/執行單位得將其報名作品作公開發表、重製、公開播送、公開展示、或重新編輯與出版作非商業用途之實施，且報名團隊不得對於上述之作品要求任何形式之報償。
- 九、 報名團隊應尊重評審委員之決定，對評審結果不得異議。
- 十、 報名截止後不得更換或新增團隊成員及學校指導老師。

十一、 為維持競賽公正性和評分需求，主辦/協辦/執行單位有權要求入圍決審隊伍和得獎隊伍提供其詳細原始設計資料文件。

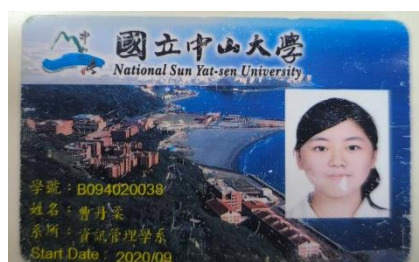
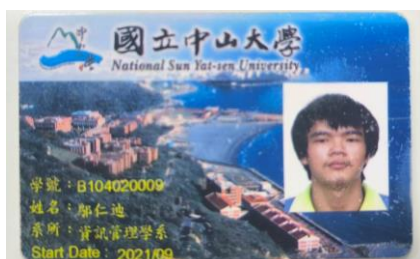
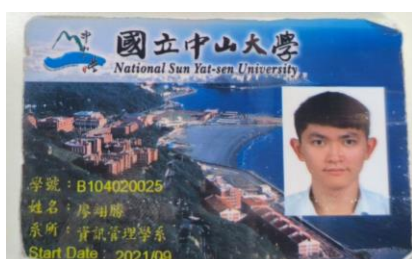
十二、 參賽者同意主辦/協辦/承辦單位得將其個人資料及相關作品納入「Tech New Stars 科技新秀大賽」資料庫中。

所有參賽者簽名：(請掃描簽名圖檔上傳)

團隊成員	姓名 (請清楚填寫或繕打)	簽名 我已完整閱讀且同意上述規定 (須由本人親筆簽名)
	危湘好	危湘好
	林子曉	林子曉
	廖翊勝	廖翊勝
	鄔仁迪	鄔仁迪
	曹丹柔	曹丹柔
	王雍翔	王雍翔

## 學生身分證明文件（隊伍成員均須提供）

### 學生證正/反面或在校證明書



# 第三屆 「Tech New Stars 科技新秀大賽」

## 申請書

組別：生成式 AI

團隊編號：2024GB0004

隊伍名稱：Recording Start

作品名稱：自然語言導向開發程式系統  
驗證與評估(CodeBRT)

## Tech New Stars 科技新秀大賽

組 別：☐智慧機器人組；☒生成式 AI 組

隊伍名稱：Recording Start

主 題：自然語言導向開發程式系統驗證與

評估(CodeBRT)

壹、作品摘要（300 字內）

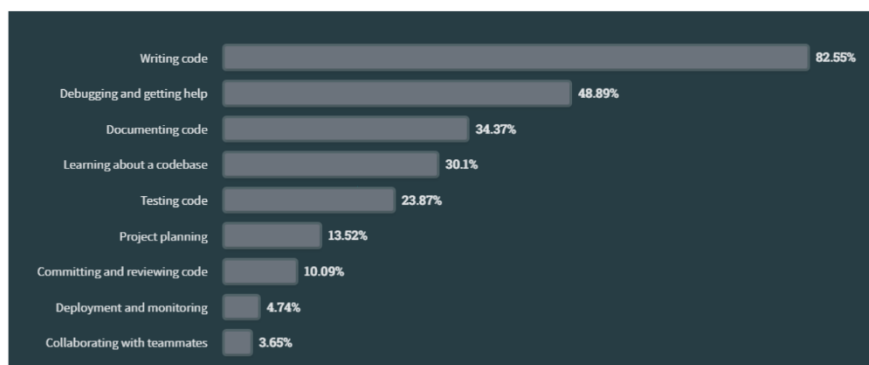
調查顯示，超過 70%的程式設計師使用生成式 AI 輔助編程，顯示其普及性。然而現有環境以程式碼為導向，缺乏高品質指示支持和專案結構統整規劃，限制了效率和質量的提升。本作品之核心在於研究自然語言導向開發（NLOD）系統，主要目標是建立一套融合自然語言處理技術的程式開發環境，並且能夠針對現有開發環境中的專案結構建議、文件自動建構、程式碼自動生成及指示品質進行改進等功能，期望提升開發效率、程式碼品質和降低程式開發的難度。

開發階段分為四個階段，每個階段都有明確的主題，並且承上啟下，第一階段目標為基礎問答功能，第二階段目標為圖像與音訊服務，第三階段目標為程式碼編輯區功能和第四階段代理人自動完成。

貳、動機

- 軟體開發已經與生成式 AI 無法脫離：根據 Survey reveals AI' s impact on the developer experience 調查收集超過 100 個北美公司員工，高達 92%受訪者 有使用生成式 AI 協助開發，並有 70 % 用戶認為這項技術給予工作優勢。

使用者利用AI協助的內容



2023 Stack Overflow Developer Survey

[1]使用者用 AI 協助撰寫程式

- 現有開發環境限制：當前的程式開發環境主要以程式碼為導向，儘管一些環境已經嵌入了生成式 AI 的輔助功能，但仍然缺乏對於專

案結構劃分和建議的統整支持，限制了開發效率與質量的進一步提升。

- 語言模型對變更的追蹤不足：語言模型生成內容大部份都需要額外修正後才能有效使用，但使用線上功能的語言模型無法自動追蹤用戶修正過後的程式碼，且易受聊天記錄干擾，導致需要重複給予修改指示，降低開發效率。
- 模型對當前開發狀態理解不足：軟體開發應該是有序建立檔案，並依照順序開發，語言模型很常只關注自己當前任務，而忽視應優先處理項目，例如： 應該在撰寫 API 前先建立模型定義。
- 指示好壞是依賴於當前開發階段：對於不同項目指示會有不同寫法，像是 Python 與 Java 開發邏輯就差異很大。即便是在同個項目中，不同組件也需要依據功能進行動態調整指示。
- 程式編寫規範無法簡單統一：大型專案由多人完成，程式編寫者有可能對公司或團體的規範有不清楚或遺漏，需要系統能夠在使用者的編寫過程中和最後繳交時檢驗程式是否完全符合團體規範。

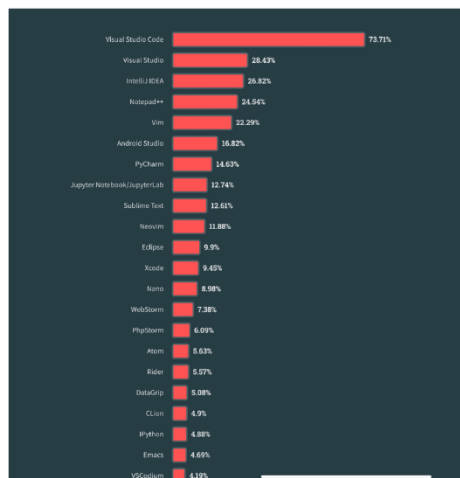
#### 參、目的

- 專案結構規劃建議：事先規劃好一個好的專案結構，可使使用者在之後維護程式碼時更有效率。
- 文件自動建構：許多專案擁有相似的結構，若有方法能自動建構專案文件，能使我們省去文件分類的時間。
- 實現語言模型對程式碼生成與用戶修改的持續追蹤：在整個開發過程中實時追蹤和理解程式碼的生成與用戶進行的修改，程式碼被修改時，可以調用設置的語言模型或分析工具去分析，並進行反饋。
- 提升模型理解當前開發狀態的系統性整理機制：有效的資料結構和演算法，可系統性地整理和保存開發過程中的關鍵資訊，使用有意義的標記，整理程式碼的功能、設計和邏輯結構。
- 程式編寫規範審核：在使用者完成程式的過程中和提交到 Github 前，由系統檢驗是否符合團體規範要求且執行順利。

#### 肆、創意描述

- 以 VSCode 插件為架構：市面上的程式生成網頁與實際使用者執行的 IDE 有一段距離，為了使用 AI 服務而來回切換視窗會浪費時間。選擇 VSCode 為開發環境是因為它是目前多數開發者使用的 IDE 之一。

- 根據根據使用者需求 step by step 的建立任務：協助程式開發或專案管理新手能夠有序完成任務。



使用VSCode 作為開發工具人數



在開發過程中是否使用AI輔助

2023 Stack Overflow  
Developer Survey

#### [1] VSCode 使用人數

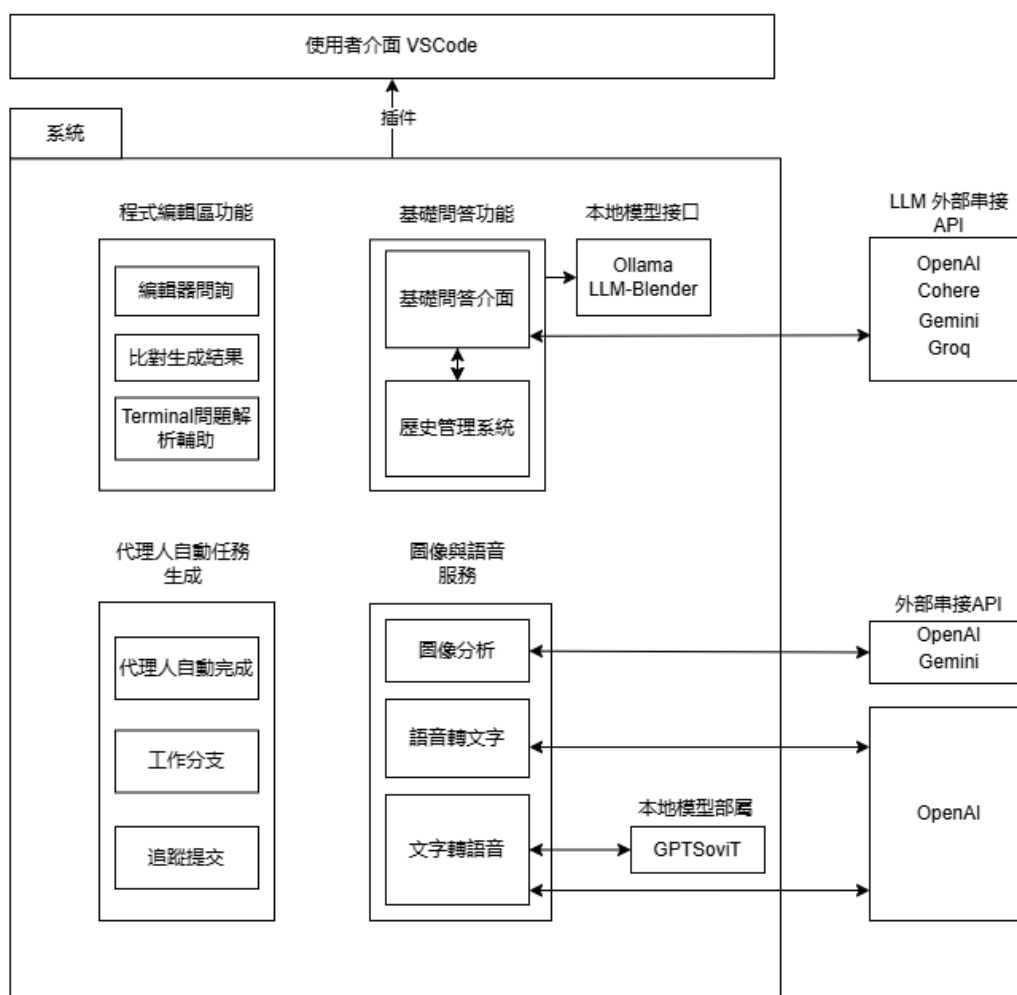
- 提供語言模型聊天的歷史記錄修改：目前網頁上的大部分 LLM 無法在網頁更改歷史聊天紀錄，LLM 在參考聊天紀錄回答時就會將前面錯誤的資訊納入回答問題的依據資料，使錯誤影響 LLM 現在的回答。所以我們提供讓使用者更改錯誤歷史紀錄的機會，避免錯誤延續。
- AI 修改前後對照：讓使用者對照 AI 修改前後的程式，了解 AI 具體修改了哪些檔案及程式碼以降低調適及排查錯誤的難度。
- 有語音及圖像辨識功能：只有少數的 IDE 有語音及圖像辨識功能。
- 集成多種語言模型的回覆：根據 LLM-BLENDER(Ensembling Large Language Models with Pairwise Ranking and Generative Fusion) [2]不同的要求和 prompt 會讓 LLM 有不同的高低成效，所以我們提供多個語言模型的回覆，讓使用者能夠自行選擇適合的應用內容。
- 在 IDE 進行跨檔案的程式生成與追蹤：目前的 IDE 是透過單一檔案給予程式生成建議。我們提供跨檔案的程式生成與追蹤，讓 LLM 的回答更貼近使用者需求。

市面上有多個程式輔助開發軟體，我們的優勢是將多個使用者還未想到的功能，以不同的架構模式組合，讓使用者更便捷快速的開發心儀的程式。也是以程式開發者的角度思考目前市面上提供的服務如何能夠更貼近程式開發流程。

## 伍、技術運用



## 系統架構



自動代理人框架：代理人自動任務完成

技術概述：這是一個多代理人對話框架，允許建立獨立的工作分支，由各個代理人負責執行任務，並由其中一個代理人總管，確保任務之間互不干擾，讓使用者能夠方便地構建 LLM 工作流程。

技術選型：本專案使用了 AutoGen 框架 (An Open-Source Programming Framework for Agentic AI) [3]，提供多樣應用構建和增強 LLM 推理與優化，涵蓋各個領域和複雜性。

技術優勢：使用 autogen 技術，可以便捷地建立多代理人對話框架，支持多樣化應用開發測試和優化 LLM 推理效能。

集成式 LLM 生成

技術概述：提供 LLM-Blender 實現集成式生成的服務。

技術選型：LLM-Blender 可以集成多個語言模型，並根據不同需求選擇最佳生成結果，確保生成程式碼的質量和多樣性。



技術優勢：集成式 LLM 生成技術通過結合多個語言模型的優勢，提供更高質量和多樣性的生成結果，滿足不同使用者的需求。

#### 實時追蹤與反饋系統

技術概述：實時追蹤和記錄使用者對生成程式碼的修改，提供持續的反饋和優化。

技術選型：開發了一個實時追蹤系統，能夠記錄使用者對生成程式碼的每次修改，並提供相應的反饋和優化建議。

技術優勢：實時追蹤與反饋系統增強了模型對使用者修改的適應性，提高了生成程式碼的實用性和使用者的開發效率。

#### 集合式 AI

技術概述：透過市面上已有的 LLM 模型 API 和本地模型介面實現生成式 AI 功能。

技術選用：整合了 OpenAI 和其他主流 LLM 模型的 API 接口，同時提供本地模型的接口，確保在網路不穩定或敏感資料需要本地處理時，依然可以使用生成式 AI 功能。

技術優勢：整合多種 LLM 模型和本地模型接口，增強了系統的靈活性和可靠性，使用戶在不同場景下都能有效利用生成式 AI。

#### 語音服務

技術概述：進行語音與文字之間的互相轉換，支援中、英、日文和自然語言處理功能。

技術選型：利用 OpenAI API 和 GPTSoviT 框架，OpenAI 提供先進的自然語言理解和生成模型，支援多種語言提供語音與文字之間的互相轉換。GPTSoviT 框架，用於文字轉語音的技術框架，可以根據提交的音訊檔生成所需的文字內容和聲音。

技術優勢：基於 OpenAI 的先進模型，能夠準確地將語音和文字互轉，也支持多種語言。模擬聲音 GPTSoviT 框架，可以給予使用者客製化的聲音選擇。

#### 程式語言

技術概述：使用 TypeScript、React.js、Node.js 建構整體系統架構。

技術選用：選擇 TypeScript 和 React.js 作為前端開發語言和框架，因其類型安全性和元件化開發方式能夠提高程式碼的可維護性和開發效率。Node.js 則作為後端服務，提供高效率的 API 介面和即時通訊功能。

## 陸、可行性及應用性

提供一個集成了 ChatGPT、Gemini 等廣泛應用的大型語言模型的 IDE 界面，使用者可以在一個直觀易懂的前端頁面上自定義 API 串接。使用者可以簡單地使用 API 串接，也可以通過 Host 導入想要使用的模型。此外，使用者可以根據自己的喜好自定義界面的顏色，包括在與語言模型對話中生成的代碼區域的顏色。

該界面允許使用者播放與語言模型對話的片段，並提供語音轉文字的功能。使用者還可以通過設定將自己喜歡的音色輸入到語音模型中，以模仿該音色。

使用者可以在與語言模型的對話中修改某個節點的聊天記錄，模型將從該節點開始生成新的對話，同時保留該節點之前的聊天內容，使用者無需重新開始一輪對話。在目前最多使用者的 IDE VS code 上開發，減少用戶學習新開發環境的成本

在一個項目或一個功能開發過程中，許多企業會要求程式碼須符合自訂的規範，每間企業的規範又不盡相同，為此我們提供企業程式需求規範客制化，可根據企業要求修改需要檢測內容並自動提出修改建議生成報表資料，像是一些企業所要求的程式格式，對不同企業所要求的檢測項目做出調整，現有市場雖然有提供程式規範檢測功能，如 Prettier，Checkstyle 等，但他們的可定制性有限，爲了更好的解決這一部分需求，我們的系統提供更有針對性的自定義檢測功能，滿足企業的程式碼規範客制化需求。

目前用戶在使用語言模型服務協助開發代碼時，需要不斷切換視窗，我們認爲這會降低使用者的開發效率及帶來不流暢的使用體驗，我們目前提供了比 Copilot 更多元的集成式服務，針對使用者提供集成式 API，發揮集成服務的優勢，將多種服務整合在一個界面中。使用者可以使用單一 API 直接訪問多種服務，包括與多個大型語言模型對話、圖片識別、語音播放等功能，盡量減少開發者開發過程的切換成本，提升效率。

開發者在使用生成式 AI 輔助開發過程中，通常都需要對修改的生成的代碼進行修改，但是現有的語言模型無法直接追蹤生成的代碼，我們的系統提供代碼的生成及修改的實時追蹤，可記錄使用者對代碼的，並調整後續生成內容，確保生成的代碼更符合使用者的實際需求。

本項目將根據使用者需求，提供合理的項目結構建議，並制定高質量的階段性工作任務指示，自動生成所需項目目錄，降低開發門檻。

## 柒、結論

在當前的開發環境中，生成式 AI 的輔助功能已逐漸成為提升專案開發效率和提升程式碼品質的重要工具。本專案考慮了目前程式開發者的需求及現有工具的局限性和不便性，以自然語言導向開發系統為核心，集成多個大

型語言模型和多媒體處理技術，提供一個強大且易用的 VS Code 插件，實現了專案結構建議、文件自動建構、程式碼自動生成和指示品質改進等功能。

本專案選擇了目前多數開發者使用的 IDE，以 VS Code 插件為架構，讓使用者得以在一個的介面中完成所有的開發工作，也避免了因切換視窗而浪費時間的問題。本專案還具備語音及圖像辨識功能，增強了多媒體處理的能力，並集成多種語言模型的回覆功能，讓使用者能夠自行選擇適合的內容應用，以增加系統的實用性及靈活性。我們還會提供集成式 API，它整合了專案中各種語言模型和功能，讓使用者得以使用單一 API 直接訪問多種服務，不僅簡化了開發流程，還能提高開發效率。

另外，本專案的自動代理人框架使用多代理人對話系統，允許創建獨立的工作分支，每個代理人負責特定任務，由總代理人負責監督和協調，確保任務之間互不干擾，讓使用者能夠更方便地建構 LLM 工作流程，以實現自動化任務完成。本專案提供語言模型聊天歷史紀錄修改的功能，避免因錯誤資訊延續而影響後續回答的問題。本專案也支持在 IDE 中進行跨檔案的程式生成與追蹤，讓生成建議更貼近使用者需求，以提升建議的準確性。並且，透過 AI 生成或修改前後程式碼對照的功能，讓使用者也能了解具體新增或修改的內容，以降低調適和排查錯誤的難度。這些功能不僅能幫助開發者的工作效率提高，更進一步提升了開發體驗。

除此之外，本專案還提供靈活的自定義程式碼檢測功能，以滿足企業對程式碼規範的特殊需求。我們觀察到業界中許多公司對於程式碼審查的工具有著實際的需求。因此，我們也額外提供了針對企業需求的程式規範客製化服務，能夠根據企業的要求調整檢測內容，並自動生成修改建議和報告。

本專案不僅提升了開發效率和程式碼品質，還大幅降低了開發門檻，幫助開發者能更快地適應不同類型的開發需求，也為使用者提供了一個高效且智能的開發環境。

## 捌、參考資料及其他

[1] <https://survey.stackoverflow.co/2023/#technology-most-popular-technologies>

[2] LLM-BLENDER(Ensembling Large Language Models with Pairwise Ranking and Generative Fusion): <https://arxiv.org/pdf/2306.02561>

[3] AutoGen (An Open-Source Programming Framework for Agentic AI) : <https://microsoft.github.io/autogen/>