**靜宜大學資訊工程學系**

**畢業專題口試系統文件規格書**

**線上編譯**

實驗室名稱：507實驗室

指導教師：謝孟諺老師

專題學生：

資工四A 411018345 林祐德

資工四A 411030371 曾偉哲

資工四B 411030460 張燿棕

1. 前言…………………………………3
2. 系統功能…………………………………4
3. 使用對象…………………………………………5
4. 使用環境…………………………………5
5. 開發工具…………………………………5
6. 相關圖片…………………………………6
7. 成本分析…………………………………8
8. 結論及未來發展…………………………………9
9. **前言**

　　在現代軟體開發中，編譯器是每個程式開發者必須掌握的重要工具之一。然而，對於很多初學者或沒有編程經驗的人來說，配置編譯環境常常是一個繁瑣且容易出錯的過程。為了解決這一問題，我們開發了一個線上Web編譯器平台，旨在幫助使用者無需繁瑣的環境設置即可快速編譯和執行程式碼，透過集成GitHub登入，即時編譯並執行的功能以及即時聊天窗的交流。所以我們希望可以提供一個簡單又便捷的入門編譯工具。

我們為了實現這一平台，我們在前端採用了Next.js技術框架，後端則基於Nest.js 開發。我們選擇這些技術的原因如下：

**為什麼選擇 Next.js作為前端框架？**

* **伺服器端渲染 (SSR)**：Next.js支援伺服器端渲染，提升網頁的加載速度與 SEO 效果，適合需要快速響應的 Web 應用。
* **內建路由**：Next.js 提供簡潔的路由系統，讓開發者能快速建立頁面結構。
* **豐富的生態系統**：作為 React 的擴展框架，Next.js 擁有豐富的資源與社群支持，易於整合各類前端工具與技術。

**為什麼選擇 Nest.js作為後端框架？**

* **模組化設計**：Nest.js採用模組化架構，便於管理與擴展應用功能。
* **強大的型別支持**：基於 TypeScript 開發，提供強型別系統，降低程式錯誤並提升開發效率。
* **與前端協作良好**：Nest.js能輕鬆整合 GraphQL 或 REST API，為前端提供穩定、高效的數據接口。
* **豐富的內建功能**：例如依賴注入、認證模組等，讓開發者能專注於業務邏輯的實現。

我們希望藉由這個簡單又便捷的入門編譯工具，為程式開發新手提供良好的起點，也讓經驗豐富的開發者能更高效地測試和驗證程式碼。

1. **系統功能**
2. 編輯器功能
   1. 語法高亮：根據所選語言自動高亮顯示代碼中的關鍵字、變量和注釋。
   2. 自動補全：在用戶編寫代碼時自動提供關鍵字和函數的建議，提升編碼效率。
   3. 代碼格式化：自動整理代碼格式，保持代碼整潔和易讀。
3. 編譯和運行功能
   1. 即時編譯：用戶編寫代碼後能立即編譯，顯示編譯結果和錯誤信息。
   2. 多語言支持：支持多種編程語言（如C/C++、Python、JavaScript等）的編譯和運行。
4. 文件管理功能
   1. 文件上傳：允許用戶上傳本地文件到線上編譯器。
   2. 項目管理：提供項目結構管理功能，支持多文件項目開發。
5. 協作功能
   1. 實時協作：多名用戶可以同時編輯同一份代碼，查看對方的修改。
   2. 即時性的討論：可以讓多個使用者即時、同步的討論與協調。
6. 安全和隱私功能
   1. 資源限制：對運行的代碼設置資源限制（如CPU時間、內存使用），防止資源濫用。
   2. 數據保護：確保用戶數據和代碼的安全，防止未經授權的訪問。
7. **使用對象**

主要使用對象為靜宜大學的學生與老師。

1. **使用環境**

當提到學習與教學的最佳化工具，線上編譯器無疑成為不可或缺的一環。透過線上編譯器，老師在授課時能監控學生的進度，迅速辨識並協助解決問題，從而打造更高效的互動式教學環境。同時，學生也能充分體驗團隊協作的樂趣，所有人可以在同一個檔案中編輯、留言與討論，彷彿置身於一個虛擬的共享工作坊。

1. **開發工具**
2. 使用技術：
3. HTML/CSS/JavaScript：用於構建線上編譯器的用戶界面。
4. Next.js :是一個基於 React 的前端框架，主要用於構建高效的 Web 應用程序。
5. React：用於構建響應式和組件化的用戶界面。
6. Nest.js：使用TypeScript的後端框架。
7. Docker：做容器化的環境設定
8. MY SQL數據庫：統整數據的工具和技術可以幫助開發者進行更有效的管理、分析和可視化操作。
9. GitHub OAuth 2.0：做 GitHub 驗證登入
10. 開發環境：
11. Visual Studio Code : 做程式碼撰寫與編譯，並且連接至主機做主機的設定。
12. Linux : 做虛擬機的架設，與統一環境。
13. Docker：在Linux中使用Docker做到容器化，可以讓我們快速的down和up。
14. **相關圖片**

一張含有 文字, 螢幕擷取畫面, 軟體, 多媒體軟體 的圖片

自動產生的描述

圖(一)、主畫面

**圖(一)**上半部紅框代表**程式碼撰寫部分**

* 可以多人連接到同一個檔案
* 目前預計支援語言 js, c, c++, python

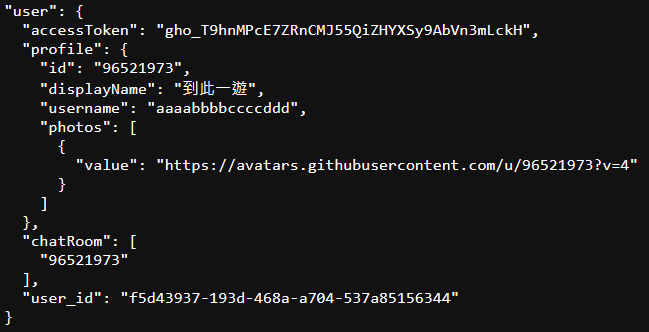
**圖(一)**下半部黃框代表**程式碼輸出部分**

*  將程式碼執行的結果輸出在下半部分

圖(二)、相關API

**圖(二)**相關API

* 採用restful api 將使用者傳入的資料儲存或回傳特定資料
* POST為新增、GET為查找、PATCH為更新、DEL為刪除



圖(三)、GitHub回傳資料

**圖(三)** **GitHub回傳資料**

* 在完成登入以後，會回傳GitHub個人資料
* user\_id是從資料庫抓下來的使用者id，而不是GitHub的

1. **成本分析**
2. 經費預算

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 項目名稱 | 說明 | 單位 | 數量 | 單價 | 小計(元) |
| 個人電腦 | 專案之進行 | 部 | 3 | 35,000 | 105,000 |
| 顯示器 | 開發環境需求 | 台 | 3 | 8,000 | 24,000 |
| 網站伺服器 | 部署與運行環境 | 部 | 1 | 50,000 | 50,000 |
| 網路設備 | 提供穩定網路環境 | 套 | 1 | 10,000 | 10,000 |
| 軟體授權費 | 編譯器及工具使用 | 套 | 1 | 15,000 | 15,000 |
| 技術人力 | 開發與維護人力 | 月 | 6 | 60,000 | 360,000 |
| 維運費用 | 伺服器托管與流量費 | 月 | 6 | 10,000 | 60,000 |

在此總計624,000元

1. **結論及未來發展**
2. 結論

我們希望可以在教學上使用一個線上編譯器做到同步、即時的網站，使我們的教學更加快速，並且保持優質的教學品質，也可以讓新生感受到Git的方便性。也希望藉由這項功能，讓新生之間可以更加方便的提出與解決問題，促進學生們的自學能力。

1. 未來發展
2. 可以增加與GitHub的聯繫，包含將專案推進使用者分支(目前有權限問題)，可以生成Git分支的圖形化介面。
3. 提升程式碼品質，增進程式效率。
4. 人工智能和機器學習的應用。
5. 智能代碼補全：利用AI技術提供更智能的代碼補全和建議，提升編碼效率。
6. 錯誤檢測和修復建議：自動檢測代碼中的錯誤並提供修復建議，減少調試時間。