

2024 Frontend Intern Homework

丹尼爾是一名工程師，他過往學一些人使用 GitHub Issue 來充當自己的部落格。他已使用 GitHub Issue 寫部落格一段時間，但發現這樣做有些明顯的缺點，不只是顯示上比較侷限，文章也比較不容易被搜尋引擎排到前面。因此他決定串接 GitHub API 並使用 React.js 開發一個網頁，讓搜尋引擎更容易尋找到他在 GitHub Issue 上寫出的文章，並調整成自己喜歡的樣式，希望熟悉前端的你能幫助他完成這個專案。

題目描述

串接 GitHub API，讓作者在登入後能夠「瀏覽」、「新增」、「更新」Blog 文章，「瀏覽」留言；非作者僅能「瀏覽」Blog 文章及「瀏覽」留言。

基本要求

- 使用 **React.js** 或基於此的框架，例如 **Next.js**
- 請使用 Git 版本控制，並將程式碼上傳至 GitHub 上，作業完成後繳交連結即可
- 請在 README 內詳細說明如何啟動專案與作業架構的設計

功能

- GitHub Login
 - 請串接 GitHub OAuth，讓使用者有權限操作 GitHub API
 - 詳見 [GitHub OAuth documentation](#)
 - 注意：在登入時需要要求正確的 `scope`
- Post Management
 - 請將 GitHub Issue 作為 Post，以 GitHub Issue 實作，並將 close Issue 視為刪除 Post
 - 你可能會需要 [GitHub Issue documentation](#) 或 [GitHub GraphQL API Explorer](#)
- User Interface
 - 列表頁
 - 第一次只能載入 10 筆
 - 每當列表滾到底部時要需要自動發送 API 請求，並載入額外 10 筆，直到沒有更多文章
 - 文章頁
 - 顯示文章內容，並正確 render 出 markdown 的內容
 - 使用者可以在此「編輯」、「刪除」
 - 新增 / 編輯文章時，可以使用 Modal 或跳轉至新的頁面操作
 - 至少需要使用 `title` 和 `body` 兩個欄位
 - 表單驗證：title 為必填，body 至少需要 30 字

加分條件

- 使用 TypeScript
- 使用 Next.js + App Router
- 調校 Web Vitals 評分
- 有處理錯誤及例外狀況 (Error Handling)

- 有部署至線上環境

評分項目

- 正確性：必須符合基本要求、功能能正常運作
- 效能：例如避免重複發送 API 請求、避免 component re-render
- 程式碼架構與品質：例如可讀性、一致性、重用性