



## 2023 Frontend Intern Homework

強尼是一名工程師，他想要幫助開發團隊更有效地管理專案。團隊已長期使用 GitHub，但他們在使用 Issue 來追蹤進度上遇到困難。因此他決定串接 GitHub API 並使用 React.js 開發一個網頁來提供更有用的專案管理工具，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，使用戶能夠「新增」、「更新」、「搜尋」Task，並能夠更新 Task 的狀態。

### 基本要求

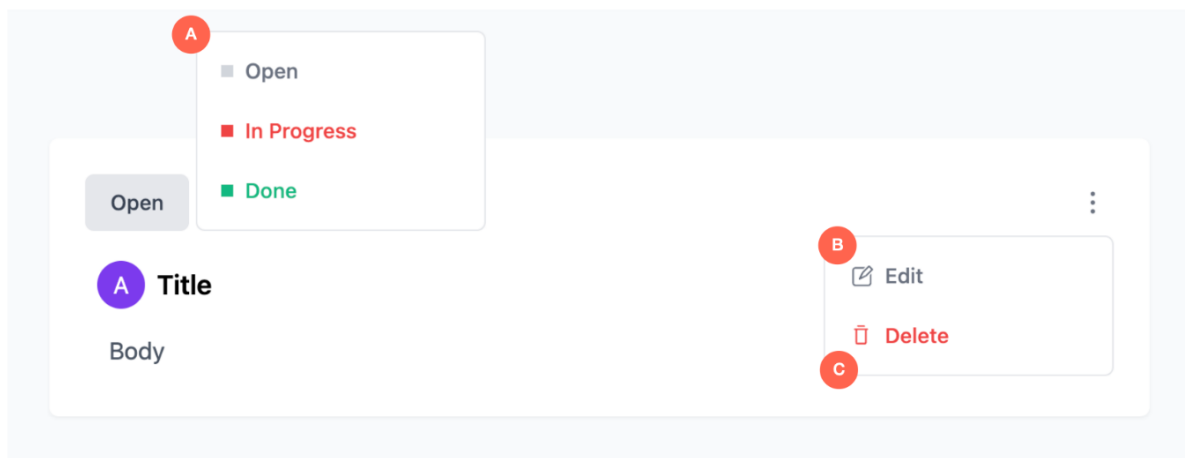
- 使用 **React.js** 或基於此的框架，例如 **Next.js**
- 不限制使用任何第三方 library
- 請使用 Git 版本控制，並將程式碼上傳至 GitHub 上，作業完成後繳交連結即可
- 請在 README 內說明如何啟動專案與作業架構的設計

### 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`
- Task Management
  - 請將 GitHub Issue 作為 Task，以 GitHub Issue 實作，並將 close Issue 視為刪除 Task
  - 你可能會需要 [GitHub Issue documentation](#) 或 [GitHub GraphQL API Explorer](#)
- Task Search

- 使用 GitHub Search API 搜尋
- 你可能會需要 [GitHub Search 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需要自動發送 API 請求，並載入額外 10 筆，直到沒有更多 Task
    - 能夠根據 Task 的狀態 (Open / In Progress / Done) 進行篩選，預設顯示所有 Task
    - 能夠根據建立的時間進行排序，預設根據建立時間從新到舊
    - 能夠根據 Task 的內容進行搜尋
  - Task 詳情頁
    - 顯示 Task 內容、狀態
    - 使用者可以在此「編輯」、「刪除」
  - 新增 / 編輯 Task 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

## 範例



- A Status - GitHub issue API (update labels, ex. Open / In Progress / Done)**
- B Edit - GitHub Issue API (update title or body)**
- C Delete - GitHub Issue API (update state to closed)**

## 加分條件

- 使用 TypeScript
- 未使用現成的 GitHub API client (如 Octokit) 串接 GitHub API

- 有處理例外狀況 (Error Handling)
- 有 deploy 至線上環境

## 評分項目

- 正確性：必須符合基本要求、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：例如易讀性、一致性、重用性