

React.js 前端工程師面試題目

專案概述

請使用 React.js 開發一個瀑布流相片牆應用，包含搜尋過濾功能。

功能要求

1. 瀑布流相片牆

- 實作響應式瀑布流布局
- 相片按不同高度錯落排列
- 支援無限滾動載入更多相片
- 在不同螢幕尺寸下自動調整欄位數量

2. 搜尋功能

- 提供搜尋框輸入關鍵字
- 根據相片作者姓名進行過濾
- 搜尋結果即時更新
- 支援清空搜尋恢復完整列表

3. 基本互動

- 載入狀態提示
- 錯誤處理和提示
- 流暢的使用者體驗

API 資訊

端點

GET <https://picsum.photos/v2/list?page={page}&limit={limit}>

參數

- `page`: 頁碼，從 1 開始
- `limit`: 每頁相片數量，建議 20-30 張

回傳資料格式

json

```
[
  {
    "id": "0",
    "author": "Alejandro Escamilla",
    "width": 5000,
    "height": 3333,
    "url": "https://unsplash.com/photos/yC-Yzbqy7PY",
    "download_url": "https://picsum.photos/id/0/5000/3333"
  }
]
```

搜尋實作說明

- 由於 API 不支援搜尋參數，請在前端實作搜尋功能
- 以 `author` 欄位作為搜尋目標
- 支援模糊搜尋（包含關鍵字即可）

技術要求

必須使用

- React.js (建議使用 Function Components + Hooks)
- 響應式設計 (支援手機、平板、桌面)
- 原生 JavaScript fetch 或 axios 進行 API 呼叫

可選技術

- TypeScript
- CSS-in-JS / Styled Components / CSS Modules
- 自訂 Hooks
- 第三方 UI 庫

實作提示

瀑布流布局

可使用以下任一方式實作：

- CSS Grid
- CSS Flexbox
- JavaScript 動態計算位置
- 第三方庫（如 react-masonry-css）

效能優化建議

- 圖片懶加載
- 搜尋防抖 (Debounce)
- 避免不必要的重新渲染

使用原始圖片比例

利用 API 回傳的原始 `width` 和 `height` 數據來計算圖片在瀑布流中的實際顯示尺寸：

```
javascript

// 根據容器寬度計算圖片高度
const displayHeight = (containerWidth / originalWidth) * originalHeight;
```

可搭配 Picsum 的動態尺寸功能：

```
https://picsum.photos/id/{id}/300 // 寬度300，高度自動按比例
```

評估重點

我們主要評估：

1. **功能完整性** - 是否實現所有要求功能
2. **程式碼品質** - 結構清晰、可讀性好
3. **專案架構** - 檔案組織和元件設計
4. **使用者體驗** - 互動流暢、載入狀態處理
5. **響應式設計** - 在不同裝置上的表現
6. **錯誤處理** - API 錯誤和異常情況處理

交付要求

1. **完整的 React 專案**，可直接執行
2. **README.md** 包含：
 - 安裝和執行指南
 - 專案說明
 - 技術選擇說明
3. **程式碼** 需包含適當的註解

祝您順利完成！如有任何問題，請隨時詢問。