

# Tabi-OS：互動式工具平台

現代雲端桌面：模組化、多工具、社群驅動

指導老師：陳玉專

專題學生：B12091202 陳英全

# 大綱

1. Tabi-OS: 互動式工具平台專題報告
2. 大綱
3. 研究動機與目標
4. 專案緣起與核心理念
5. 主要目標功能規劃
6. 專案總覽與核心特色
7. 技術棧
8. 系統架構與資料夾結構
9. 資料夾結構 (1/4): 公用資源與主要應用 Apps
10. 資料夾結構 (2/4): 共用元件 Components
11. 資料夾結構 (3/4): 核心配置與基礎設施
12. 資料夾結構 (4/4): 全域資源與 Next.js App
13. 主要功能與使用者體驗
14. 研究方法與開發流程
15. 核心程式片段範例
16. 優點與挑戰
17. 研究成果展示 (UI 截圖)
18. 未來展望
19. 如何貢獻 (簡要)
20. 感謝您的關注與參與！

# 研究動機與目標

- 現代數位工具普遍存在分散、效率低落的問題，導致用戶需在多平台間頻繁切換，造成操作不便與困擾。
- Tabi-OS 旨在打造**一開放、模組化且可擴充**的雲端桌面平台。
- **核心目標：**促進協作與創新，同時降低大眾使用與開發者貢獻的門檻。

# 專案緣起與核心理念

- **緣起：** 現有工具平台常因封閉或付費模式，限制了用戶的自訂彈性與社群貢獻機會。Tabi-OS 受多工具協作網站啟發，致力於提供更佳解決方案。
- **核心理念：**
  - **模組化設計：** 各應用程式獨立存放於 `/src/apps/`，實現高內聚、低耦合。
  - **自動註冊機制：** 新應用可透過 `/src/config/appRegistry.ts` 自動整合，無需修改全域程式碼，開發者得以專注單一 App 開發。
  - **開放架構：** 鼓勵開源精神、促進社群協作與持續創新，打造以社群為核心的現代雲端桌面。

# 主要目標功能規劃

- **平台目標：** 建立穩健、高效能且可擴充的雲端桌面環境，基於 Next.js、React 及 Supabase。
- **核心功能：**
  - 整合多種實用工具 (如書籤、音樂播放器、計算機、記事本、天氣查詢等)。
  - App Store：提供使用者探索、安裝、移除應用程式的介面，實現個人化桌面。
  - 多視窗系統：模擬真實作業系統的操作邏輯與體驗。
- **技術優勢：**
  - SSR (伺服器端渲染) 提升初始載入效能與 SEO，並結合 `manifest.json` 與結構化 metadata 優化。
  - 全面支援無障礙 (a11y) 與多語系 (i18n)，並採用極簡美學設計原則。

# 專案總覽與核心特色

- **Tabi-OS:** 一個模組化、用戶驅動的桌面平台，提供流線型工具組。

- **核心特色：**

- 每個 app 獨立於 `/src/apps/`，自動註冊，易於擴充
- App Store 探索/安裝/移除應用程式
- 多視窗系統，模擬真實桌面操作
- SSR (伺服器端渲染) 提升效能與 SEO
- 無障礙設計 (a11y) 與多語系支援 (i18n)
- 重視效能、極簡設計與卓越使用者體驗
- 開源精神，鼓勵社群參與及貢獻

# 技術棧

- **核心框架:** Next.js (v15.x), React (v19.x)
- **語言:** TypeScript (v5.x)
- **狀態管理:** Jotai (原子化、可擴展)
- **樣式:** Tailwind CSS (v4.x)
- **UI 元件:** Radix UI, shadcn/ui
- **後端/資料庫:** Supabase (PostgreSQL, Auth, RLS)
- **套件管理/執行環境:** Bun (v1.x)
- **程式碼品質:** ESLint, Husky, Commitlint
- **其他:** Docker, Cloudflare, Google reCAPTCHA, Vercel

# 系統架構與資料夾結構

## 架構概述

- **前後端分離：**
  - 前端 (UI/UX, 狀態): Next.js, React, Jotai
  - 後端 (資料, 認證): Supabase
- **模組化設計：**每個應用程式 ( /src/apps/ ) 獨立，高內聚低耦合。
- **自動發現機制：**新應用無需修改全域程式碼即可註冊 ( /src/config/appRegistry.ts )。

# 資料夾結構 (1/4): 公用資源與主要應用 Apps

```
1 public/                                # 公用資源根目錄（靜態檔案）
2   └ backgrounds/...                      # 背景圖片資源
3   └ icons/...                           # 網站與應用的圖示資源
4   └ metadata/...                         # 網站的中繼資料（如 Open Graph、manifest 等）
5     └ sounds/...                          # 聲音資源（如提示音效等）
6
7 src/                                   
8   └ apps/                                # 所有用戶應用程式（每個 app 一個資料夾）
9     └ clock/                             # 範例：時鐘應用
10    └ components/                       # 應用程式專屬 React 元件
11    └ atoms/                            # 應用程式專屬 Jotai atoms（狀態）
12    └ hooks/                            # 應用程式專屬自定義 hooks
13    └ config.ts                         # 應用程式設定（icon、預設大小等）
14    └ types.ts                          # 應用程式專屬 TypeScript 型別
15    └ index.ts                          # 應用程式註冊主入口
16    └ README.md                         # （選用）應用程式說明文件
17   └ calculator/                       # 同上結構
18   └ ...                                # 其他應用程式
```

# 資料夾結構 (2/4): 共用元件 Components

```
1 src/  
2 | ... (public/, apps/)  
3 |- components/          # 共用 UI 元件  
4 | |- ui/                 # Shadcn、Radix 等基礎 UI  
5 | |- layout/             # 佈局元件 (Window、Taskbar、Sidebar 等)  
6 | |- appstore/           # 應用程式商店/探索元件  
7 | |- auth/               # 認證 UI (登入表單等)  
8 | |- desktop/            # 桌面圖示及相關元件  
9 | |- debug/              # 除錯工具/元件  
10 |- shared/              # 共用元件 (如 LoadingSpinner)  
11 |   |- providers/        # Context 提供者 (Theme, Auth, Jotai 等)
```

# 資料夾結構 (3/4): 核心配置與基礎設施

```
1  src/
2  | ... (public/, apps/, components/)
3  |- application/          # 全域應用程式邏輯
4  |   |- atoms/            # 全域 Jotai atoms (window、auth 等)
5  |   |- hooks/            # 全域自定義 hooks
6  |- config/               # 全域設定
7  |   |- appRegistry.ts    # 所有可用 app 的註冊中心
8  |   |- ...                # 其他全域設定檔
9  |- infrastructure/      # 基礎建設層
10 |  |- lib/                # 外部函式庫、SDK (如 Supabase client)
11 |  |- utils/              # 工具函式 (如 report.ts、localStorage helpers)
```

# 資料夾結構 (4/4): 全域資源與 Next.js App

```
1  src/
2  | ... (public/, apps/, components/, core logic/)
3  |   locales/           # 多語系檔案 (en.ts、zh-TW.ts 等)
4  |   styles/            # 全域樣式 (Tailwind CSS 等)
5  |   types/             # 共用型別/介面
6  |   utils/             # 工具函式
7  |   app/               # Next.js app 目錄 (路由、佈局、API)
8  |     | [locale]/       #   多語系頁面/佈局
9  |     | api/            #   API 路由
```

# 主要功能與使用者體驗

- **模組化 App 系統：**易於新增、管理、移除工具。
- **自動發現 App：**簡化開發者貢獻流程。
- **多視窗互動：**類真實桌面體驗，提高生產力。
- **App Store：**提供集中的工具探索與管理平台。
- **持久化狀態：**如回報冷卻時間，透過 localStorage 記憶。
- **無障礙 (a11y) 與響應式設計：**兼顧所有使用者。
- **雲端同步 (Supabase)：**支援認證、使用者偏好設定等。

# 研究方法與開發流程

## 研究方法：

- 文獻參考與現有平台分析
- 採用現代 Web 技術 (React, Next.js, Supabase, Bun, TypeScript)
- 敏捷開發，持續實作與反覆測試
- CI/CD 自動化部署流程
- 重視用戶回饋，驅動產品優化
- 強調社群參與和貢獻

## 開發流程：

1. 需求分析與目標設定
2. 架構設計與技術選型
3. 前後端開發與整合
4. UI/UX 設計與無障礙測試
5. 部署、測試與持續優化
6. 用戶測試與回饋收集
7. 文件撰寫與成果展示

# 核心程式片段範例

package.json (部分)

```
{  
  "scripts": {  
    "dev": "next dev",  
    "build": "bun run lint && next build",  
    "start": "next start",  
    "lint": "next lint --fix",  
    "prepare": "husky"  
  },  
  "dependencies": {  
    "next": "15.x",  
    "react": "19.x",  
    "@supabase/supabase-js": "^2.0.0",  
    "jotai": "^2.x",  
    "tailwindcss": "^4.x"  
    // ... 其他依賴  
  }  
}
```

此片段展示專案的關鍵依賴與核心腳本，反映了所選用的技術棧與開發流程中的自動化配置 (如 Husky 預提交檢查)。

# 優點與挑戰

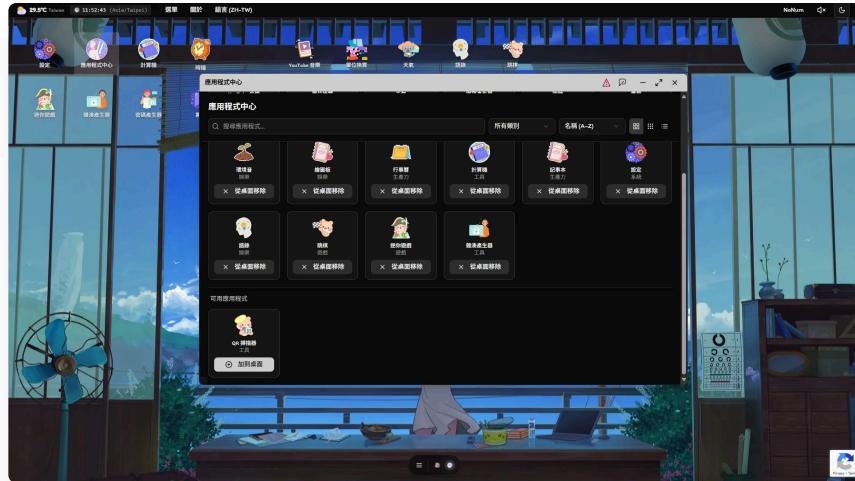
## 優點：

- 現代技術棧，高效能 (SSR)
- 模組化易擴充、易維護
- 自動 App 發現機制，簡化貢獻
- 支援多語系與無障礙設計
- 互動式、類桌面使用者體驗
- 開源且易於社群貢獻

## 挑戰與取捨：

- SSR 增加部署與除錯複雜度
- 多視窗及 App 管理的狀態複雜性
- 雲端同步的即時性與衝突處理
- 行動裝置 UX 需持續優化與適配
- 初始載入體積 (Bundle Size) 控管
- 新貢獻者對專案架構的學習曲線

# 研究成果展示 (UI 截圖)



展示多視窗桌面系統、App Store、工具應用等核心介面。

# 未來展望

- **功能擴展：**

- 支援更多第三方應用程式與插件系統
- 新增通知中心、日曆、即時聊天等核心應用
- 儀表板個人化自訂功能

- **體驗優化：**

- 強化雲端同步機制與可能的離線支援
- 行動裝置使用者體驗深度優化與適配

- **生態建設：**

- 擴展 API 接口，支援外部自動化與深度整合
- 持續優化系統效能、安全性與穩定性
- 擴大開發者社群，完善貢獻文件、教學與流程

# 如何貢獻 (簡要)

- **新增 App:**

- 於 `/src/apps/` 建立新資料夾 (建議 kebab-case 命名)
- 需包含 `index.ts` (主出口), `config.ts` (視窗配置), `README.md` (說明)
- App 將透過自動發現機制註冊至系統

- **翻譯:**

- 平台層級翻譯: `/src/locales/`
- 應用程式翻譯: `/src/apps/appName/locales/`

- **其他:** 歡迎透過 GitHub Issues 回報錯誤、提出功能建議、或協助改進文件。

更詳細的貢獻指南與規範，請參閱專案根目錄下的 `CONTRIBUTING.md` 檔案。

# 感謝您的關注與參與！

Tabi-OS 期待與您一同成長。

[專案 GitHub](#)