

Flowney – 口袋中的財務顧問

黃鉅堯

111403538 資管二B

matchaskill@gmail.com

【摘要】 本系統旨在開發 Flowney 大語言模型智能記帳軟體的競賽展示網頁版本，目的在於快速地讓評審知道實際系統的特色及運作模式，根據使用者選擇不同的數據調整，連接後端大語言模型進行不同結果的財務建議輸出。前端使用 Nuxt.js 和 Tailwind CSS 進行開發，後端使用 FastAPI 接收前端請求，大語言模型的部分使用 OpenAI 的 GPT-3.5-turbo 模型，而系統環境使用 Docker 進行容器化包裝開發。

【關鍵詞】 生成式 AI、財務、網頁開發

1. 動機與目的

由主計總處於 111 年 4 月 29 號公佈的國富統計報告可以發現，台灣家庭貧富差距正日益擴大當中，已來到近 67 倍，同時貧富差距越大則精神疾病患病率越高，其差距約為 35 倍，並且美國心理學會的研究也指出有 78% 的美國人對錢感到焦慮。團隊也發現，台灣的義務教育與高等教育環境中，除商管專業外，並未教授財管與金融等相關知識，導致國人畢業對於財務管理能力的不足。因此，或許無法改變經濟結構的現狀，但我們可以開發針對家庭與個人的財務管理 APP，讓處於財富平均數及中位數之下的國人，亦可獲得財務管理服務的支援，紀錄、知悉、調整自己的財務狀況，達成生涯目標並降低心理負擔。

2. 系統架構

本系統架構圖如圖 1 所示，在前端開發中，我們選擇了 Nuxt.js 作為主要框架。Nuxt.js 不僅提供了服務端渲染 (SSR) 和靜態網站生成 (SSG) 的功能，還能顯著提升頁面的加載速度和 SEO 性能。為了確保用戶界面的美觀和響應性，我們採用了 Tailwind CSS。這一工具不僅能讓設計變得簡潔且高度可定制，還能幫助我們快速構建出色的響應式網頁。

後端方面，我們使用 FastAPI 來接收和處理來自前端的請求。FastAPI 是一個高效的 API 框架，支持異步編程，這使得我們能夠大幅提升後端的處理性能和響應速度，從而更好地滿足系統的需求。

在大語言模型的選擇上，我們使用了 OpenAI 的 GPT-3.5-turbo 模型。這一模型具備強大的自然語言處理能力，能夠根據使用者提供的數據進行深入的分析，並生成高質量的財務建議。無論數據如何變化，GPT-3.5-turbo 都能靈活應對，提供準確且有用的建議，顯示出其強大的智能分析能力。

為了保證系統在不同環境中的一致性和可移植性，我們使用 Docker 進行容器化開發。Docker 技術使我們能夠將整個系統打包成獨立的容器，確保系統在各種環境中都能穩定運行。這不僅簡化了部署流程，還提升了開發效率，使我們能夠更快地迭代和更新系統。

總體而言，新系統架構分為四個主要區塊：前端、後端、大語言模型和系統環境。前端使用 Nuxt.js 和 Tailwind CSS 來提供友好的用戶界面和流暢的互動體驗；後端通過 FastAPI 來高效處理請求；大語言模型部分則由 OpenAI 的 GPT-3.5-turbo 配合專業領域知識的提示工程來生成智能財務建議；整個系統通過 Docker 進行容器化包裝，保證其一致性和穩定性。

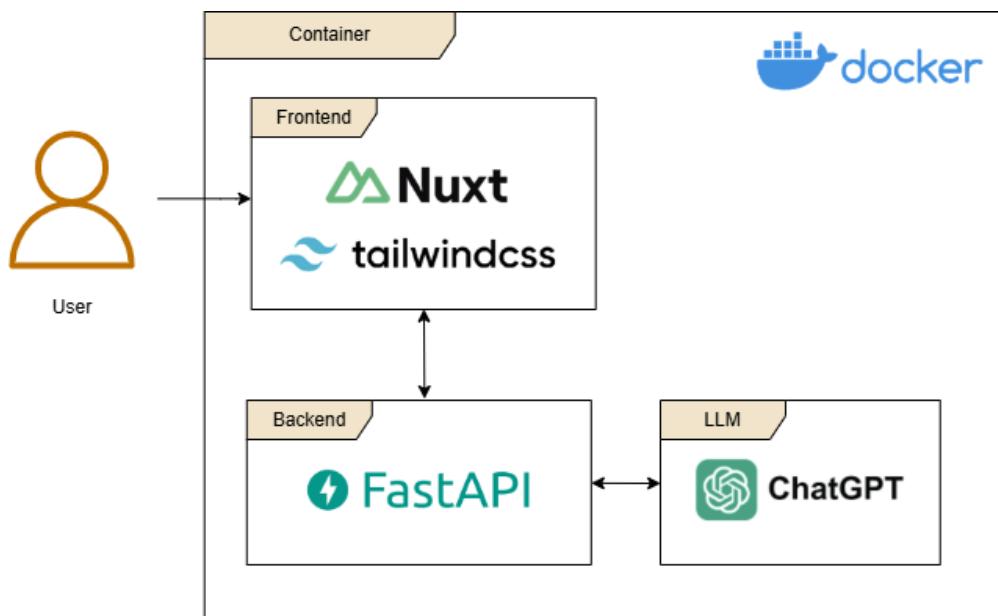


圖 1、系統架構圖

3. 各頁面功能及特色

以下介紹系統各個頁面之功能及特色，動態展示請參考錄影展示檔案。

3.1. 主畫面

此頁面為主畫面，如圖 2 所示，畫面上有閃爍黃字特效，使用者可以點擊上方藍色按鈕會有刪除文字並加入新文字的特效，頁面間的切換

也採用 slide-out-in 令轉換變得滑順。



圖 2、主畫面

3.2. 使用者資料設定

此頁面為使用者資料設定，由於此系統設計要點在於能夠快速演示效果，故只開放兩個選擇給使用者進行選取，如圖 3 所示，此頁面再進行切換時會將使用者選取的參數作為 Query 傳遞至其他頁面。

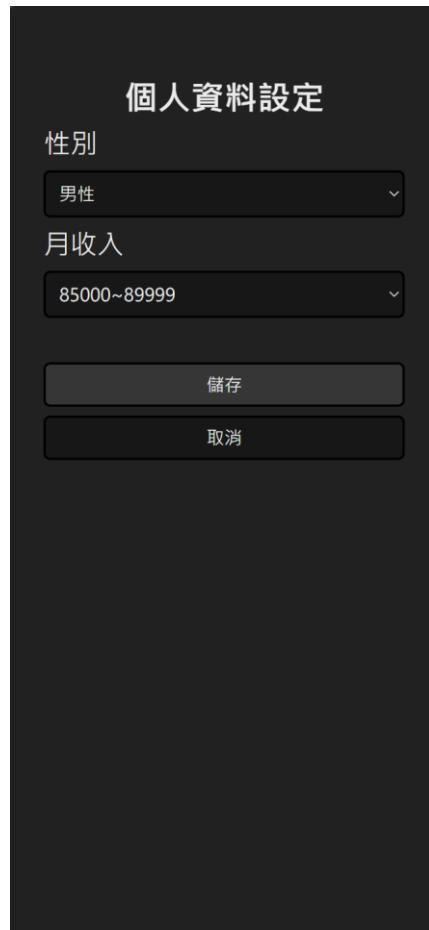


圖 3、使用者資料設定

3.3. 自動記錄帳目確認

此頁面為展示自動記帳之紀錄，可讓使用者進行左滑或右滑進行帳目確認，此頁特色為其拖拉動畫及效果，如圖 4 所示，其呈現方式為堆疊之卡牌效果，可進行層疊釋放。



圖 4、自動帳目紀錄確認

3.4. 帳目確認 & 財務資料分析

此兩頁面為過場頁面，如圖 5 所示，目的在於給予使用者心理暗示，令其認為給予的資料是經過專業運算處理後才得出最後之結論，讓使用者有更高的意願相信我們給出的結果的專業度。

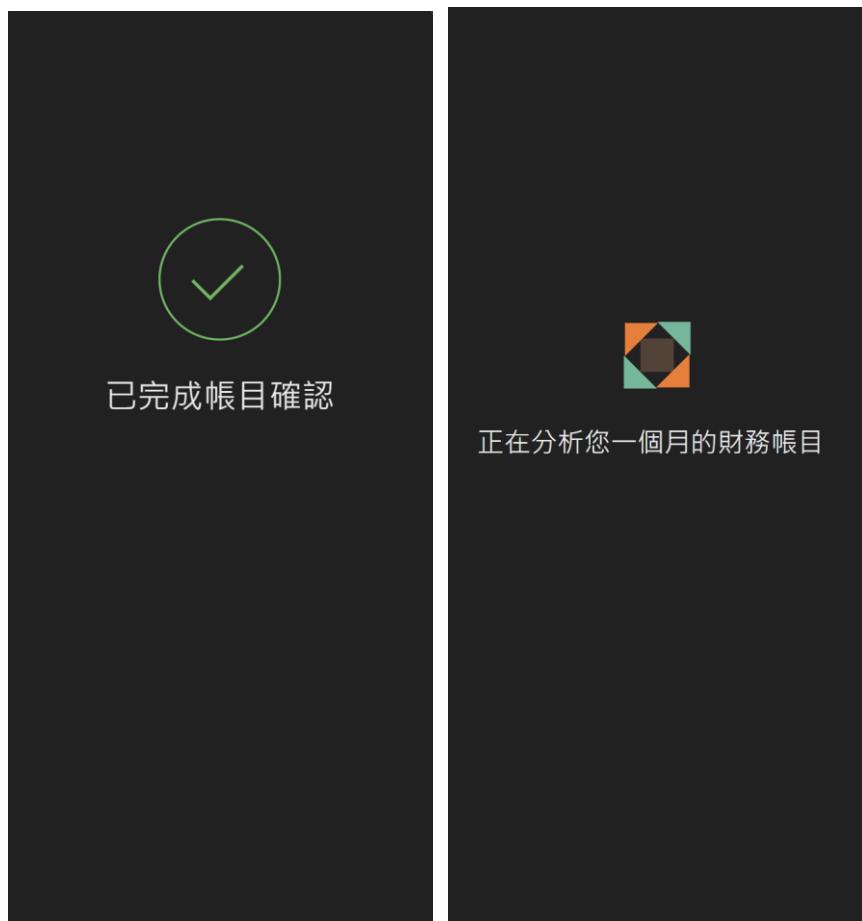


圖 5、帳目確認 & 財務資料分析

3.5. 帳目分析結果

此頁面為本系統最大特色，其目的在顯示財務建議結果，如圖 6 所示，後端根據使用者選擇資料配合經過專業領域知識調整的提示工程進行財務建議生成，在此頁面會根據前面使用者資料設定頁面得到的資料將其傳輸至後端，後端會另外呼叫撰寫好的大語言模型函數。

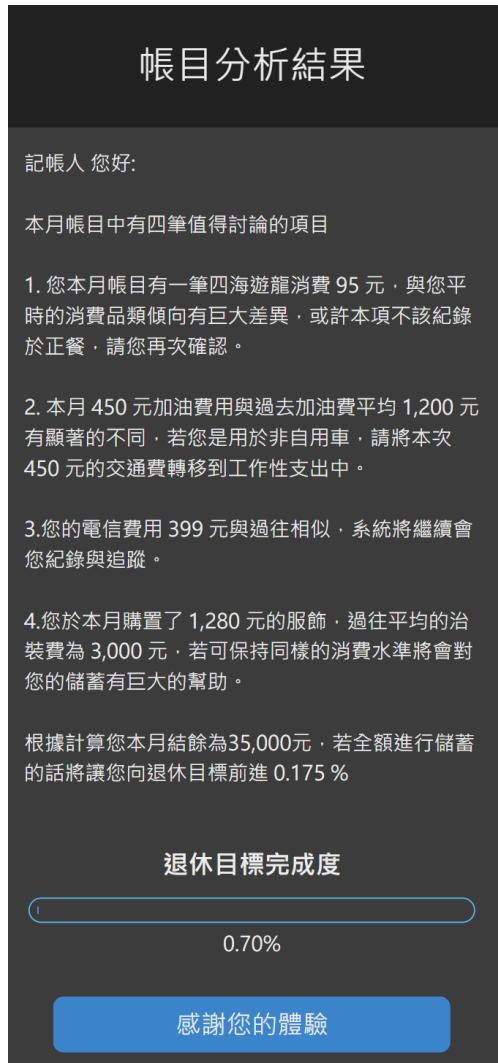


圖 6、帳目分析結果

4. 特殊技術說明

4.1. 前端

前端使用 CSS 的 keyframes 製作動畫效果，利用 fetch 向後端連接獲取資料，利用 Nuxt.js 的 router 進行頁面轉換。

4.2. 後端

將程式進行模組化包裝，從後端程式直接導入。生成式 AI 串接 OpenAI 的 gpt-3.5-turbo，配合經過專人知識驗證的 System Prompt 進行微調，確保輸出結果格式符合範文。

5. 補充資料

GitHub: [OG-Matcha/Flowney-Demo \(github.com\)](https://github.com/OG-Matcha/Flowney-Demo)

商業展示效果: "[Let me introduce you to Flowney.](#)" - YouTube