

財務管理應用專案

一、目的

本專案開發一個個人財務分析系統，透過後端讀取用戶的財務數據 (CSV 格式)，並利用人工智慧 (AI) 技術自動生成個人化的財務報告。報告將包含使用者的財務現況分析、近期收支趨勢，以及具體可行的短期和中長期財務建議，以協助使用者進行財務決策。

二、系統架構與技術

1. 前端

- React.js：建構前端介面，包括交易列表、Dashboard 與報告生成頁面。
- React Router：負責多頁面路由管理。
- Bootstrap：提供快速且美觀的一致化 UI 風格。
- Axios：用於前端向後端 API 發送和接收數據請求。
- Chart.js + react-chartjs-2：用圖表方式直觀呈現收入、支出與分類數據。

2. 後端

- Node.js + Express.js：開發 RESTful API，負責前端請求及與 MySQL 資料庫互動。

- MySQL：資料儲存，表包含 id, type, category, amount, note, date 等欄位。
- Flask：使用 Python 的 Flask 框架建立 RESTful API，提供 /generate_report 端點，用於讀取數據、進行統計計算與調用 AI 模型生成報告。
- flask-cors：解決前端跨域存取後端服務的問題。

3. 人工智慧模型整合

- 使用 Hugging Face 提供的指令型 AI 模型 Falcon-7B-Instruct，生成自然語言財務分析。
- Prompt 工程：依據真實數據構造 prompt，清晰表達需求，如財務趨勢與未來建議。
- 容錯設計：若模型無法正常載入，將自動切換至較小規模的模型以確保服務穩定。

三、功能介紹

1. 交易數據管理

- 新增交易：透過前端表單將交易數據新增至 MySQL 資料庫。
- 交易查詢：以表格形式提供交易列表，允許動態排序與搜尋。

- 修改與刪除交易資料。

2. Dashboard 與視覺化

- 呈現財務數據，如收入、支出的比例 (圓餅圖)、分類統計 (長條圖)。
- Dashboard 使用 API 即時獲取並更新數據。

3. AI 報告生成

- 後端讀取 CSV 交易數據，計算各種統計值 (例如收入支出總額、分類彙總)。
- 透過構造 prompt，呼叫 Hugging Face 模型生成完整的財務分析。
- 報告結果以 JSON 格式回傳給前端。

四、開發過程遇到的問題與解決方式

1. 前後端 API 連接問題：

- 問題描述：前後端分開運作導致無法正常串接 API。
- 解決方法：在 React 前端的 package.json 中設定 proxy 到正確後端服務位置，確保 API 路徑設定正確。

2. 埠號衝突：

- 問題描述：前端和後端預設使用同一埠號 (3000)，導致啟動錯誤。

- 解決方法：將後端服務改用其他埠號 (例如 5000)，並調整前端 proxy 設定。
3. 套件版本衝突：
- 問題描述：path-to-regexp 版本不兼容導致 Express 路由異常。
 - 解決方法：指定安裝特定舊版本 (0.1.7)，並清除後重新安裝 node_modules。
4. 報告未提供明確建議：
- 問題描述：模型輸出只描述現狀，缺乏短期與中長期明確具體建議。
 - 解決方法：改善 Prompt 工程，明確要求短期與中長期各提出三項具體建議。
5. 模型回覆數據數值不準確：
- 問題描述：模型回覆內容出現數值誇張或不一致。
 - 解決方法：降低生成參數 temperature，提高模型輸出穩定性。

五、系統流程

透過前端 React 網頁介面：

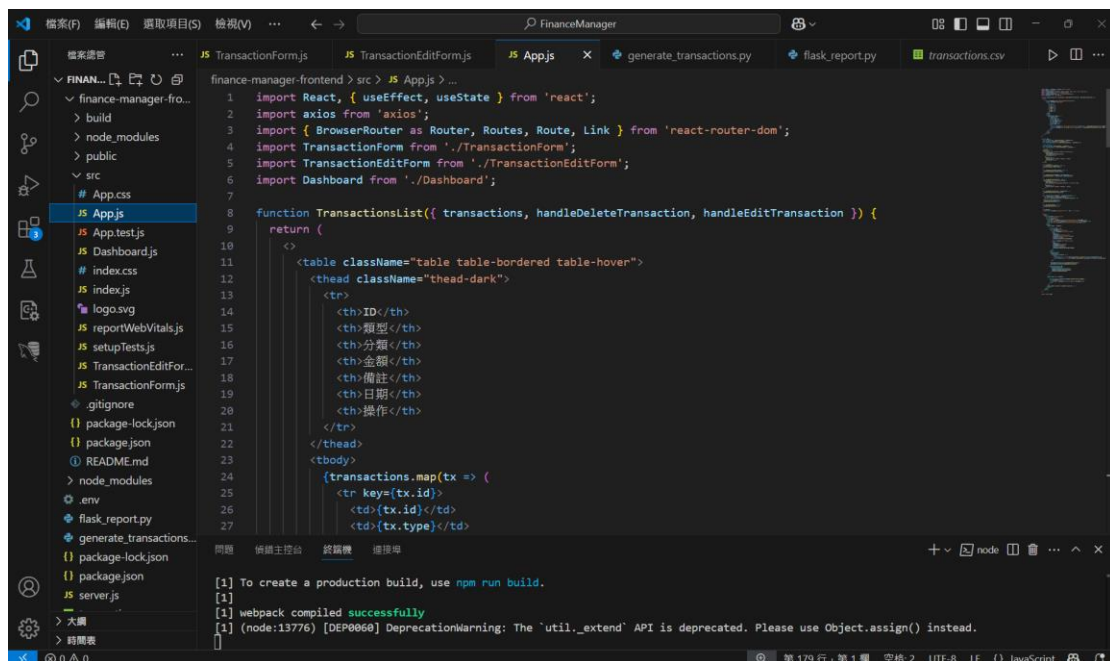
1. 新增、編輯、刪除交易數據。
2. 查看交易列表並使用 Dashboard 視覺化圖表觀察財務狀況。

3. 點擊生成報告按鈕後，前端透過 Axios 呼叫後端 Flask API。
4. 後端自動從 CSV 或資料庫取出數據，進行分析統計。
5. 組合 prompt 並透過 Hugging Face API 呼叫 AI 模型生成完整報告。
6. Flask API 將生成內容以 JSON 回傳給前端。
7. 清晰呈現報告，包含近期收入支出趨勢，提供短期及中長期共六點財務規劃建議。

六、代碼展示

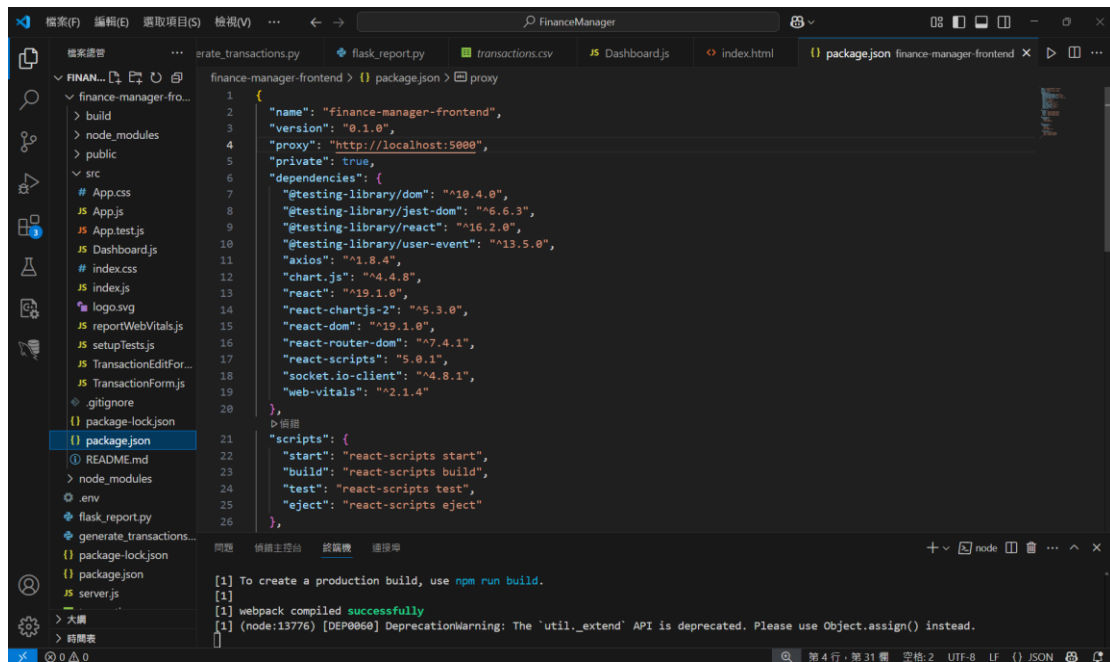
前端：

1. App.js：為控制前端主頁面顯示部分

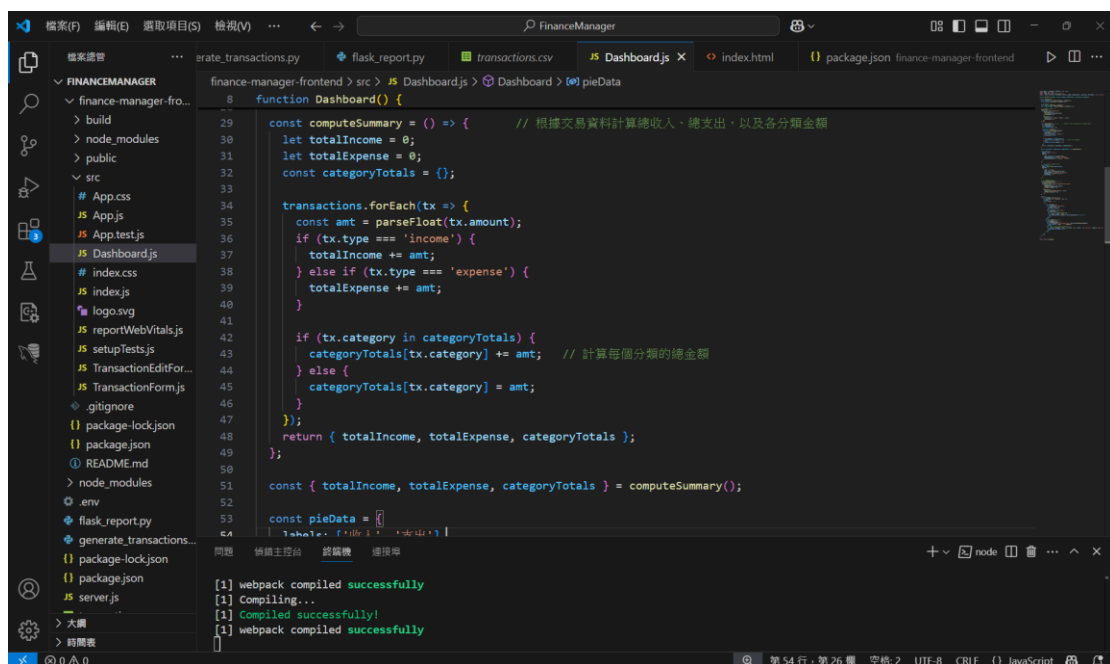


```
1 import React, { useEffect, useState } from 'react';
2 import axios from 'axios';
3 import { BrowserRouter as Router, Routes, Route, Link } from 'react-router-dom';
4 import TransactionForm from './TransactionForm';
5 import TransactionEditForm from './TransactionEditForm';
6 import Dashboard from './Dashboard';
7
8 function TransactionsList({ transactions, handleDeleteTransaction, handleEditTransaction }) {
9   return (
10     <div>
11       <table className="table table-bordered table-hover">
12         <thead className="thead-dark">
13           <tr>
14             <th>ID</th>
15             <th>類型</th>
16             <th>分類</th>
17             <th>金額</th>
18             <th>備註</th>
19             <th>日期</th>
20             <th>操作</th>
21           </tr>
22         </thead>
23         <tbody>
24           <{transactions.map(tx => (
25             <tr key={tx.id}>
26               <td>{tx.id}</td>
27               <td>{tx.type}</td>
```

2. package.json：為前端之 package.json，其中 proxy 設置為 port5000

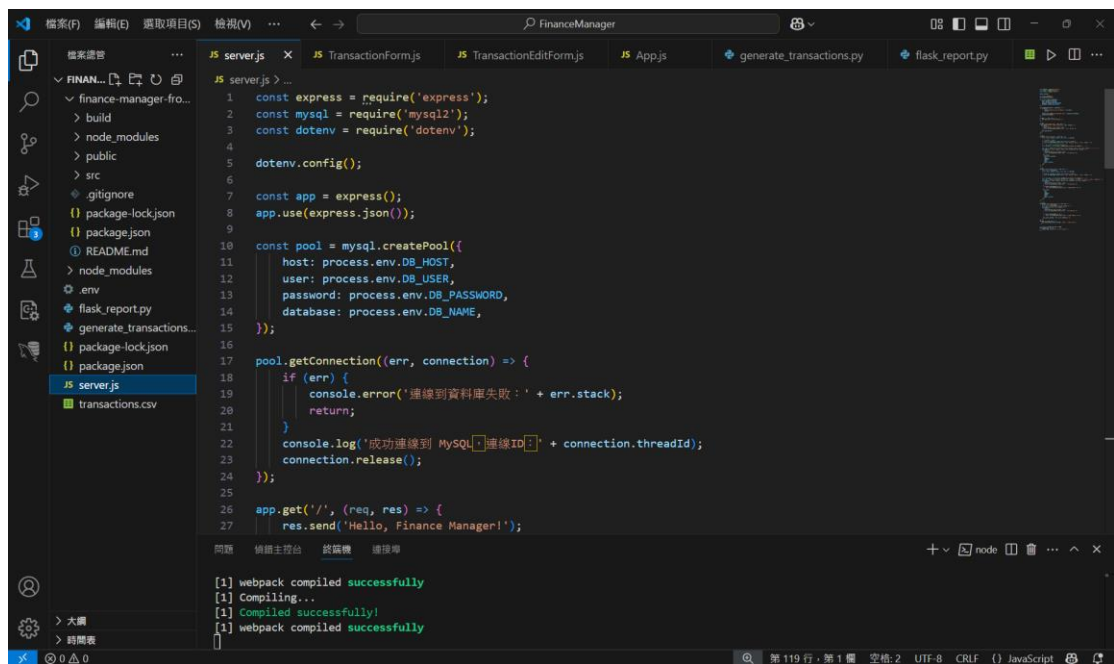


3. DashBoard.js：計算與可視化圖表繪製

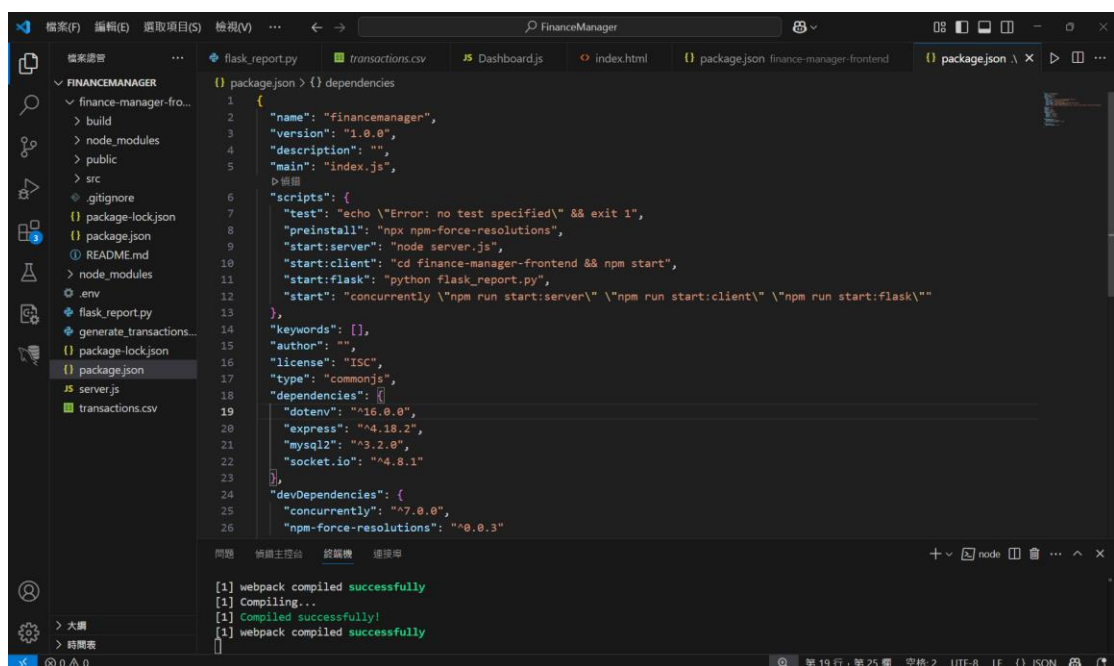


後端

4. server.js 管理後端與建立路由

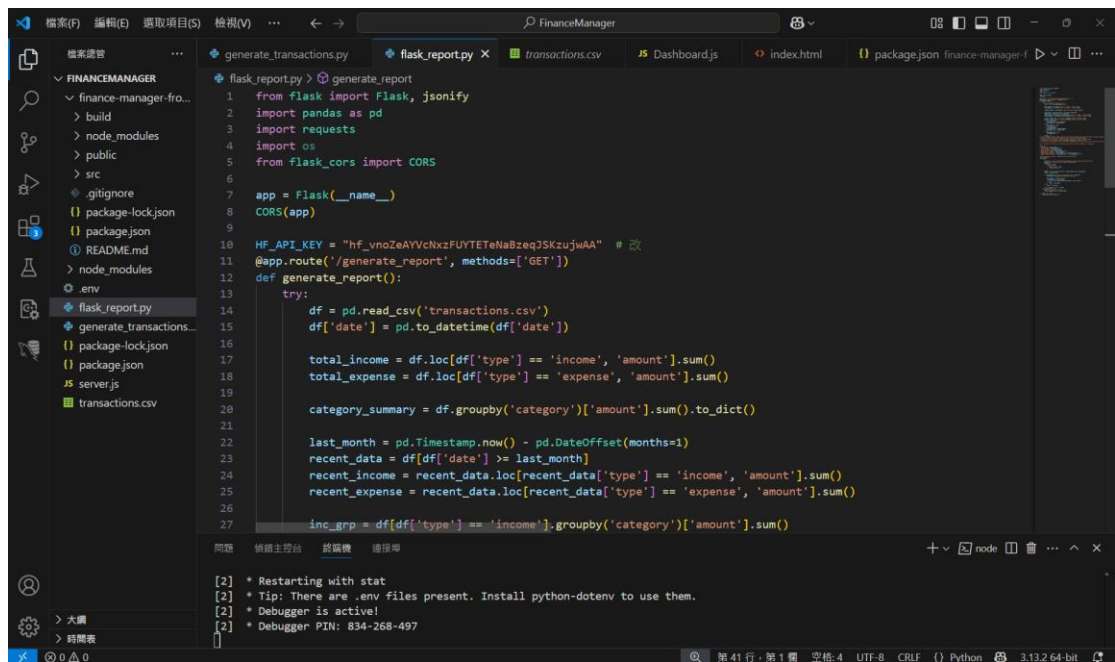


5. package.json：管理後端之 package.json，其中包含各項指令



AIGC-flask_report.py

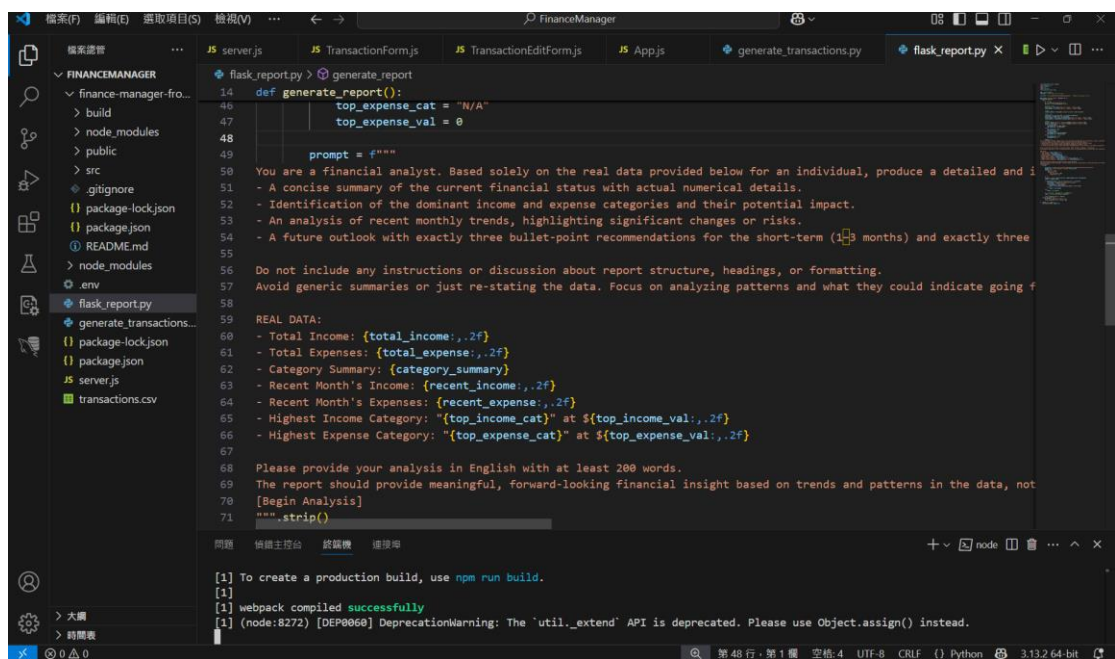
1. 讀取 csv 檔案，並對其做整合計算



```
flask_report.py > generate_report
1 from flask import Flask, jsonify
2 import pandas as pd
3 import requests
4 import os
5 from flask_cors import CORS
6
7 app = Flask(__name__)
8 CORS(app)
9
10 HF_API_KEY = "hf_vnoZeAYVcNkzFUYTETeNaBzeqJ5KzujwAA" # 改
11 @app.route('/generate_report', methods=['GET'])
12 def generate_report():
13     try:
14         df = pd.read_csv('transactions.csv')
15         df['date'] = pd.to_datetime(df['date'])
16
17         total_income = df.loc[df['type'] == 'income', 'amount'].sum()
18         total_expense = df.loc[df['type'] == 'expense', 'amount'].sum()
19
20         category_summary = df.groupby('category')['amount'].sum().to_dict()
21
22         last_month = pd.Timestamp.now() - pd.DateOffset(months=1)
23         recent_data = df[df['date'] >= last_month]
24         recent_income = recent_data.loc[recent_data['type'] == 'income', 'amount'].sum()
25         recent_expense = recent_data.loc[recent_data['type'] == 'expense', 'amount'].sum()
26
27         inc_grp = df[df['type'] == 'income'].groupby('category')['amount'].sum()
28
29     except Exception as e:
30         return jsonify({'error': str(e)})
31     return jsonify({
32         'total_income': total_income,
33         'total_expense': total_expense,
34         'category_summary': category_summary,
35         'recent_income': recent_income,
36         'recent_expense': recent_expense,
37         'inc_grp': inc_grp.to_dict()
38     })
39
40 if __name__ == '__main__':
41     app.run(debug=True)
```

[2] * Restarting with stat
[2] * Tip: There are .env files present. Install python-dotenv to use them.
[2] * Debugger is active!
[2] * Debugger PIN: 834-268-497

2. 模型依照 prompt 執行適當之自然語言輸出



```
flask_report.py > generate_report
14 def generate_report():
15     top_expense_cat = "N/A"
16     top_expense_val = 0
17
18     prompt = f"""
19     You are a financial analyst. Based solely on the real data provided below for an individual, produce a detailed and i
20     - A concise summary of the current financial status with actual numerical details.
21     - Identification of the dominant income and expense categories and their potential impact.
22     - An analysis of recent monthly trends, highlighting significant changes or risks.
23     - A future outlook with exactly three bullet-point recommendations for the short-term (1-3 months) and exactly three
24
25     Do not include any instructions or discussion about report structure, headings, or formatting.
26     Avoid generic summaries or just re-stating the data. Focus on analyzing patterns and what they could indicate going f
27
28     REAL DATA:
29     - Total Income: {total_income:,.2f}
30     - Total Expenses: {total_expense:,.2f}
31     - Category Summary: {category_summary}
32     - Recent Month's Income: {recent_income:,.2f}
33     - Recent Month's Expenses: {recent_expense:,.2f}
34     - Highest Income Category: "{top_income_cat}" at ${top_income_val:,.2f}
35     - Highest Expense Category: "{top_expense_cat}" at ${top_expense_val:,.2f}
36
37     Please provide your analysis in English with at least 200 words.
38     The report should provide meaningful, forward-looking financial insight based on trends and patterns in the data, not
39     [Begin Analysis]
40     """
41     return jsonify({'report': prompt.strip()})
42
43 if __name__ == '__main__':
44     app.run(debug=True)
```

[1] To create a production build, use `npm run build`.
[1]
[1] webpack compiled successfully
[1] (node:8272) [DEP0060] DeprecationWarning: The 'util._extend' API is deprecated. Please use Object.assign() instead.

七、實作展示

1. 前端主頁面

類型: 輸入 income 或 expense

分類: 輸入分類

ID	類型	分類	金額	備註	日期	操作
1	income	薪資	30000.00	2月薪水	2025-04-01T16:00:00.000Z	<div>編輯刪除</div>
2	expense	房租	10000.00	3月租金	2025-03-04T16:00:00.000Z	<div>編輯刪除</div>
4	expense	交通	120.00	捷運月票	2025-03-06T16:00:00.000Z	<div>編輯刪除</div>
5	income	投資回報	5000.00	股票紅利	2025-03-09T16:00:00.000Z	<div>編輯刪除</div>
6	income	獎金	5000.00	季度獎金	2025-04-29T16:00:00.000Z	<div>編輯刪除</div>
8	income	獎金	500.00	零用錢	2025-04-30T16:00:00.000Z	<div>編輯刪除</div>
10	income	其他	10.00	路上撿到錢	2025-04-20T16:00:00.000Z	<div>編輯刪除</div>

2. 編輯頁面

6	income	獎金	5000.00	季度獎金	2025-04-29T16:00:00.000Z	<div>編輯刪除</div>
8	income	獎金	500.00	零用錢	2025-04-30T16:00:00.000Z	<div>編輯刪除</div>
10	income	其他	10.00	路上撿到錢	2025-04-20T16:00:00.000Z	<div>編輯刪除</div>

編輯交易 (ID: 1)

類型:

income

分類:

薪資

金額:

30000.00

備註:

2月薪水

日期:

年 / 月 / 日 -- --:--

儲存

取消

3. 新增頁面

交易列表Dashboard新增交易

新增交易

類型:

income

分類:

學習

金額:

500

備註:

日期:

年/月/日 -- --:--

新增交易

返回列表

4. 查詢頁面

交易列表Dashboard新增交易

類型:

分類:

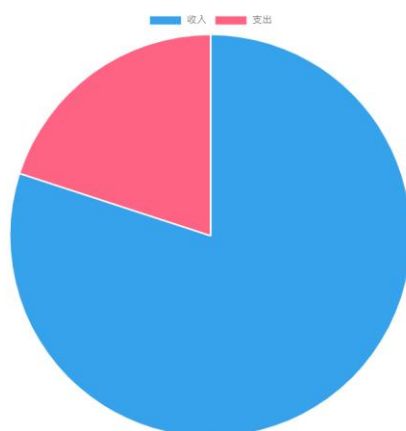
ID	類型	分類	金額	備註	日期	操作
1	income	薪資	30000.00	2月薪水	2025-04-01T16:00:00.000Z	<div>編輯刪除</div>
5	income	投資回報	5000.00	股票紅利	2025-03-09T16:00:00.000Z	<div>編輯刪除</div>
6	income	獎金	5000.00	季度獎金	2025-04-29T16:00:00.000Z	<div>編輯刪除</div>
8	income	獎金	500.00	零用錢	2025-04-30T16:00:00.000Z	<div>編輯刪除</div>

5. 可視化圖表頁面

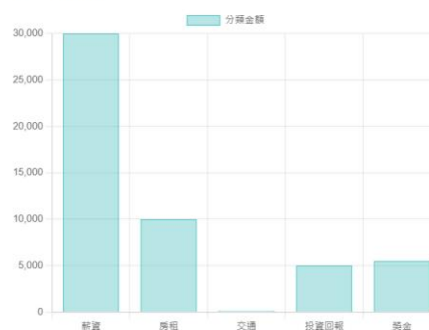
交易列表 Dashboard 新增交易

Dashboard 儀表板

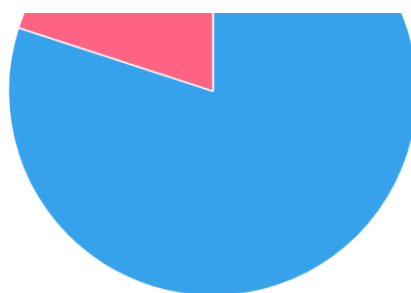
收入 vs 支出



各分類統計



6. 依照資料自動生成報告



自動生成財務報告

生成報告

The current financial status of the individual is characterized by a healthy income and expense structure. The individual's salary category accounts for the highest income, contributing \$1,598,390.68 to the overall income. This indicates a stable and reliable source of income. The expenses are spread across various categories, with the 'Others' category accounting for the highest expenditure at \$1,384,473.75. This suggests that the individual has a diverse range of financial obligations and responsibilities. Looking at the recent monthly trends, there have been fluctuations in income and expenses. The individual's salary has remained relatively stable, but expenses have increased by \$1,022,721.01 over the past month. This may indicate a rise in discretionary spending or an increase in fixed expenses. The individual should consider these trends and make adjustments to their budget accordingly. In the short-term outlook, the individual should focus on maintaining their current financial structure and reducing unnecessary expenses. This will help to ensure that their income remains stable and their overall financial health is preserved. In the medium-to-long term, the individual should consider diversifying their income streams and exploring opportunities for growth. This may involve investing in assets such as stocks or bonds, or exploring new business ventures. Overall, the individual's financial health is in a stable and positive position. By focusing on their current income and expense structure, the individual can maintain their financial stability and ensure that their long-term goals are met.

七、未來展望

1. 擴展數據來源：整合即時市場數據及第三方 API，提高財務分析的全面性。

2. 模型進一步調整與微調：透過金融專業資料集進行模型微調，提高報告的專業性與實用性。
3. 實現複合模型分析：嘗試使用多個模型分階段處理財務分析，先進行初步分析再進行內容優化。
4. 加強前端互動功能：開放用戶設定參數、預覽與下載報告等功能，增進用戶體驗。

七、結語

此專案透過整合前後端開發技術與人工智慧模型，建立一個能實際運作的個人財務管理分析系統。