

4

# HW#4 A Google-Map Full-Stack Application

## ! Copyright and Ownership Notice

關於這個 notion page 以及其包含的所有 pages (以下統稱「本 page」) · 其著作權與所有權是屬於台大電機系黃鐘揚教授以及其所開設的網路服務程式設計課程之所有助教 (以下統稱「作者們」) 所擁有。

基於推廣技術與服務大眾之精神，本 page 採取 MIT 授權，即任何人皆可因任何目的使用、複製、修改、發布、與再散佈本 page，包括商業用途。不需簽署協議，也不需支付授權費，惟任何的使用與散佈必須包含本著作權與所有權聲明。如有違反本著作權與所有權聲明，本 page 之作者們仍保有相關的法律追溯權利。詳細關於 MIT 授權的規範請見 [OSI 官網](#)。

本 page 之內容以「現狀」提供 (provided "as-is")，不附帶任何保證。對於本 page 之內容如有任何疑問或是建議，請來信  
[eewebprogramming@googlegroups.com](mailto:eewebprogramming@googlegroups.com).

The copyright and ownership of this Notion page and all pages it contains (hereinafter referred to as "this page") belongs to Professor Chung-Yang Huang of Department of Electrical Engineering, National Taiwan University, and all teaching assistants of this Web Programming course that he teaches.

In the spirit of promoting technology and serving the public, this page adopts the MIT License, meaning anyone can use, copy, modify, publish, and redistribute this page for any purpose, including commercial use. No agreement needs to be signed, and no licensing fee needs to be paid, provided that any distribution must include this copyright and ownership notice. In case of violation of this copyright and ownership notice, the authors of this page still retain the relevant legal right to take retroactive action. For detailed regulations regarding the MIT License, please see the [OSI official website](#).

The content of this page is provided "as-is", without any warranty. If you have any questions or suggestions regarding the content of this page, please email [eewebprogramming@googlegroups.com](mailto:eewebprogramming@googlegroups.com).



[Web Programming Home](#)

## 作業說明

✓ 強調：前端與後端都必須各自有 `.env` 與 `.env.example`！兩邊都要！兩邊都要！兩邊都要！（重要所以說三次）

- `.env`：本機實際設定（不要 commit）
- `.env.example`：提供欄位範例與說明（必須 commit）

請實作一個「地圖功能導向」的全端應用，採用 前後端分離架構（React + Node/Express），並整合 Google Maps API。

主題不限，但必須與「地圖 / 地點 / 座標」強相關，並完成登入、資料庫、基本 CRUD (Create、Read、Update、Delete) 與至少一項地圖互動。

目標：練習 前後端串接、本地帳密登入與資料庫整合、以及 Google Maps API 的實作。

不需部署，但架構需清晰、程式乾淨，並能完整重現測試流程。

## 可選主題（舉例，請自定一款）

- 店家/景點探索：以地圖搜尋與收藏店家或景點，建立自選清單與評分備註。
- 活動/事件地圖：建立活動（名稱、時間、地點），讓使用者查找並加入行程。
- 跑步/健走路線記錄：標記路線與里程，記錄完成日期與心得。
- 社群公益地圖：標記回收點/捐贈站/志工集合點，提供開放時間與類別。
- 租屋/二手物件分佈：發布含地點的物件資訊，可依距離或關鍵字篩選。

請避免僅做「靜態標記」；需有至少一種資料型態的 CRUD 與與地圖的雙向互動（例如：地圖點選產生地點表單、列表點選在地圖上定位）。

## 開發規格

### ◆ 前端

- 技術棧：React + TypeScript（建議使用 Vite 建置）
- 主要套件：React Router (前端 routing)、Axios (與後端的 HTTP 溝通)
- UI 框架：Material UI / Ant Design / Shadcn / TailwindCSS (擇一或混用)
- Google Maps SDK：使用 Google Maps JavaScript API 處理地圖顯示與互動

- **最低要求**

- 地圖載入與基本操作（縮放、拖曳）
- 可「搜尋」或「標記」地點（任一即可）
- 使用者登入後才能針對地點表單之類的資料進行新增/編輯/刪除（以頁面/按鈕狀態反映）

- ◆ **後端**

- 技術棧：Node.js + Express（建議 TypeScript）
- RESTful API：至少包含
  - `/auth`（註冊、登入、登出）
  - 一到兩個自定義資源（例如 `/locations`、`/events`、`/posts`、`/items`...）具備 CRUD
- Google Maps 伺服器端整合：至少串接 Geocoding 或 Places 或 Directions 任一項（依主題選擇最合理者）
- 資料庫：使用 SQLite（也可選 MongoDB 或 PostgreSQL）
  - 至少儲存「使用者登入資訊」或「主要資源資料」其中之一（建議兩者皆存）

👉 **效能評估與優化**：建議寫一些 monitors 來評估前後端在處理 Google Maps requests 時的效能，必要時做一些優化。

## 登入與安全性要求

- 帳號欄位需包含 email/username + password（其一或兩者皆可）
- 密碼必須以雜湊方式儲存（例：`bcrypt` 或 `argon2`）
- 使用 JWT 或 Session + Cookie 任一機制（請於 README 說明）
- `.env` 檔不得上傳，並需提供 `.env.example`

- 後端 CORS 設定需允許：

```
<http://localhost:5173> <http://127.0.0.1:5173>
```

**⚠** 請留意，這是你前端 Vite App 的 URL。如果你因為任何因素導致你的前端的 port 不是 5173 (可能會是 5174, 517\*, 3000, etc)，請重新確保你的前端是開在 5173，或者是修改這個設定。

- 所有輸入需驗證 (email 格式、密碼長度、必填欄位、數值/日期型態等)
- 錯誤回傳需包含正確狀態碼與訊息 (如 400/401/403/404/422/500)
- 權限控管：
  - 未登入者不可操作受保護資源
  - 登入的使用者僅能修改/刪除自己的資料

## Google Maps API 設定指南

**⚠** 開通 Google Maps API 需要綁定付款方式 (信用卡或 debit card)。Google 提供新帳號首 90 天或 \$300 美金免費額度，足以完成本作業。

**⚠** 如果不希望在免費期限結束或是額度使用完畢之後被 Google 收費的同學，請務必記得在那之前取消訂閱。如因為自己的疏忽而造成扣款，我們恕不負責。

### 一、建立 Google Cloud 專案

1. 前往 Google Cloud Console，建立新專案 (例如 [map-enhanced-app](#))
2. 確認目前選擇的專案正確

### 二、啟用必要 API (依前後端分工)

- Geocoding API / Places API / Directions API → 建立「Server Key」
- Maps JavaScript API (地圖顯示與互動) → 建立「Browser Key」

### 三、建立與設定 API Key

#### 前端 Key ( Browser Key )

- 限制類型：HTTP 網域

- 允許清單：

```
<http://localhost:5173/*> <http://127.0.0.1:5173/*>
```

⚠ 請留意，這是你前端 Vite App 的 URL。如果你因為任何因素導致你的前端的 port 不是 5173 (可能會是 5174, 517\*, 3000, etc)，請重新確保你的前端是開在 5173，或者是修改這個設定。

- 啟用 API：Maps JavaScript API
- `.env` 範例：

```
# frontend/.env.example VITE_GOOGLE_MAPS_JS_KEY=YOUR_BROWSER_KEY # Maps  
JavaScript API (Browser Key) VITE_API_BASE_URL=http://localhost:3000 #  
後端 API 位址
```

## 後端 Key ( Server Key )

- 限制類型：IP 位址 ( Note: 由於目前 app 尚未 deploy, app 是跑在本地端，故可暫時無 IP 限制，但需在 README 標註安全風險 )
- **服務啟用規則 ( 統一規格 )**：為了確保到時候同學們可以順利 review 彼此的作業，所以即使你的 app 不會完全使用到 **Geocoding, Places, Directions** 這三種 APIs，但還是請你使用 **同一把 Server Key 同時啟用下列三項服務**，以便同學互相執行與助教批改時可以使用自己的同一把金鑰來執行你的 app：
  - Geocoding API
  - Places API
  - Directions API

- **.env** 範例：

```
# backend/.env.example PORT=3000 CORS_ORIGINS=http://localhost:5173,http://127.0.0.1:5173 DATABASE_URL=file:./dev.db GOOGLE_MAPS_SERVER_KEY=YOUR_SERVER_KEY # 已啟用 Geocoding/Places/Directions
```

**⚠** 請留意，這是你前端 Vite App 的 URL。如果你因為任何因素導致你的前端的 port 不是 5173 (可能會是 5174, 517\*, 3000, etc)，請重新確保你的前端是開在 5173，或者是修改這個設定。

## 系統功能模組與最低要求

模組	功能要求
Auth ( 登入系統 )	註冊、登入、登出；帳號唯一；密碼雜湊；錯誤提示（帳號重複、密碼錯誤等）
地圖互動 ( Maps )	地圖載入；「搜尋」或「標記」地點任一；點選地圖或搜尋結果可回填表單欄位（地址/座標）
核心資源 ( 自定義 Domain )	至少 1 種資料型態（例：店家、活動、路線、物件...）具備 建立/讀取/更新/刪除；每筆資料需含 名稱、地點或座標、可選的時間或分類。核心資源必須持久化在資料庫（SQLite/MongoDB/PostgreSQL 其一），不得只存於前端 state 或伺服器記憶體陣列，需要確保前端重啟之後已經記錄的資料還在。
伺服器端 Google 服務	至少使用 Geocoding / Places / Directions 之一：如地址轉座標、依關鍵字/範圍查找地點、計算路徑/距離
( 選做 ) 擴充功能	任選一項：收藏/評分、提醒（前端模擬）、上傳/連結圖片、RWD、社群分享、離線暫存等

## API 與資料欄位建議 (通用、非強制)

- 路由設計：`/auth/*`、`/api/<your-resource>/*` (依主題命名，如  
`/api/events`、`/api/locations`)
  - 常見欄位 (可依主題調整)：  
`id`, `title/name`, `description`, `latitude`, `longitude`, `address`, `category`,  
`startTime/endTime` 或 `date`, `createdBy`, `createdAt/updatedAt`
  - 查詢參數：`?q=keyword&category=...&radius=...&lat=...&lng=...&from=...&to=...`
- 

## 開發建議流程

階段	任務	驗收方式
1	完成前端頁面 (先用假資料)	前端可操作、頁面導覽正確
2	實作後端 + SQLite + Auth (含密碼雜湊)	以 <code>curl</code> 或 <code>.http</code> 測試 API
3	串接 Google APIs ( Geocoding / Places / Directions 擇一或多 )	驗回傳結果 (記錄範例與說明)
4	前端串接後端 ( Axios )	完整流程可運作 (登入後可 CRUD)
5	文件撰寫 ( README + <code>.env.example</code> )	從零可重現

---

## 專案繳交說明

1. 將所有程式放在：

`wp1141/hw4/`

## 2. 於 deadline 前 push 到 main :

請務必寫好 .gitignore, 不要上傳不必要的檔案，例如：node\_modules, .env, logs 等

```
git add hw4 git commit -m "<your message>" git push
```

## 3. README.md 必須包含：

- 專案簡介與功能清單（說明你的主題與使用情境）
- 架構圖（可手繪或簡圖）
- 前後端啟動步驟（`npm run dev` / `npm run start`）
- `.env.example`
- 後端 API 一覽與至少 5 個 `curl` 範例（含授權流程範例）
- 已知問題與未來改進方向
- （若使用 Server Key 無 IP 限制）安全性風險說明

## 補充：批改時請替換成你自己的金鑰（務必閱讀）

同一把 server key 來啟用後端的服務，為了統一後端金鑰設定、方便彼此在本機互跑專案與助教批改：

1. **統一啟用**：如前面規定所述，請各位使用同一把 Server Key 來同時啟用 Geocoding / Places / Directions 三項服務（即使你的主題未全用到），以利 reviewer 與助教在本機批改你的 app。
2. **批改必要提醒**：你的 `.env.example` 與 README 必須明確提示評分者「需置換為自己的金鑰後再啟動」，否則地圖功能將無法運作。
3. **金鑰的 placeholder**：可在 README 提供占位符與範例指令，協助評分者了解應該要填入哪些金鑰（但請勿填寫真值）：

```
# 於專案根目錄建立/編輯 .env (請評分者自行貼上金鑰) # frontend/.env VITE
_GOOGLE_MAPS_JS_KEY=YOUR_BROWSER_KEY # backend/.env GOOGLE_MAPS_SERVER_
KEY=YOUR_SERVER_KEY # 啟用 Geocoding/Places/Directions
```

可簡單示範：將「YOUR\_BROWSER\_KEY / YOUR\_SERVER\_KEY」替換為自己的金鑰後再啟動。請勿在版本庫中提供任何有效金鑰。

# 補充：Chat History 與敏感資訊處理（務必特別留意）

為了協助檢視你的開發歷程，同時確保金鑰與個資不外洩，請遵守：

## 1) 匯出 Cursor 聊天紀錄

- 將本次作業相關的 Cursor chat history 匯出並放在：

wp1141/hw4/chat-history/

- 檔名可自訂，但請保留原始時間或主題，方便索引。



但如果您的 chat history 檔案太過龐大 (e.g. > 1MB)，請做出適當的刪減，留下 prompt, 工作項目，結論內容... 等重要資訊，然後刪除其他細節。

## 2) 由 Cursor 自行淨化（先輸出，再清理）

- 匯出後，請再用 Cursor 審閱 [wp1141/hw4/chat-history/](#)，主動要求它移除/遮罩所有敏感資訊，包含：

- API Key / Secret / Token
- 內網位址、Cookie、Session、資料庫連線字串
- 個資 (email、電話、學號...)

- 可直接貼給 Cursor 的指令型描述（範例）：

「請讀取 wp1141/hw4/chat-history/ 的所有檔案，找出並移除或遮罩所有可能的敏感資訊（例如 API Key、Secrets、Tokens、Cookies、DB 連線字串、內網位址、個資）。請給我清理後的版本，並列出你移除的遮罩的項目類型與檔名。」

- 若 Cursor 回傳修訂內容，請覆蓋原檔或另存 [-redacted](#)，上傳前僅保留已清理版本。

## 3) 關於 [.env](#) 與金鑰暴露風險

- Cursor 預設讀不到 [.env](#)。讀不到時可能會下類似 [cat](#) 的指令嘗試讀檔。
  - 你可自行斟酌是否讓 Cursor 存取你的 Key (一般不建議)。
  - 無論如何，就算 Cursor 在對話中看到或引用了你的 Key，務必在 chat history 中移除或遮罩，不要把任何有效金鑰一併上傳 (如上項說明)。
- 請確認 [.env](#) 已在 [.gitignore](#)；並提供 [.env.example](#) 方便評測。

#### 4) 上傳前自我安全檢查清單

- `wp1141/hw4/chat-history/` 已存在，且為清理過的版本
- chat history 中沒有明碼出現的：API Key / Secret / Token / Cookie / DB 連線字串 / 個資
- `.env` 未被提交；`.env.example` 有被提交且內容完整（但不含真值）
- README 已註記：若後端 Server Key 未限制 IP 的安全風險
- 提交前以關鍵字做全域搜尋：  
尋：`key=`、`AIza`、`sk-`、`token`、`secret`、`cookie`、`password`、`Bearer` ... 等

#### 5) 建議遮罩格式

- 金鑰/憑證：保留開頭 4 碼 + `***` + 結尾 4 碼（例：`AIza****9XyZ`）
- 個資：以 `**` 取代個人識別段落（例：`s*****@ntu.edu.tw`）

重要：不要將任何有效金鑰、密碼、Cookie、Token、個資，連同 chat history 一起上傳。若不確定是否敏感，寧可遮罩。

#### 評分標準（建議比例）

項目	比例
功能完整與使用體驗	60%
加強版程式品質（含安全特性）	30%