1\_foundations/4\_lab4.ipynb 功能規格（Functional Specification）說明：

**📘 功能規格：使用 OpenAI API 並整合工具通報與使用者紀錄系統**

**1. 套件與模組匯入**

* 使用了以下 Python 模組：
  + dotenv：讀取 .env 檔案中的環境變數
  + openai：與 OpenAI API 通訊
  + json、os、requests：標準 I/O 與 HTTP 處理
  + pypdf.PdfReader：處理 PDF 檔案（尚未使用）
  + gradio：提供 Web UI（後續用於介面啟動）

**2. 初始化設定**

* 使用 load\_dotenv() 從 .env 檔讀取變數，如 OpenAI 金鑰與 Pushover 通知帳號資訊。
* 初始化 OpenAI 物件以便後續 API 呼叫。

**3. Pushover 即時通知整合**

* 從環境變數中取得 PUSHOVER\_USER 與 PUSHOVER\_TOKEN
* 定義 push(message) 函式，透過 Pushover API 傳送通知訊息。

**4. 使用者紀錄功能**

* record\_user\_details(email, name, notes)：當使用者表達聯絡意願時，記錄其 email、姓名與備註，並發送通知。
* record\_unknown\_question(question)：記錄任何 Chatbot 無法回答的問題並發送通知。

**5. OpenAI 工具函數定義（JSON格式）**

* 兩個 JSON 結構用來向 OpenAI 傳遞工具定義：
  + ***record\_user\_details\_json***：讓 LLM 可以主動呼叫 record\_user\_details 函數（例如用戶留 email）
  + ***record\_unknown\_question\_json***：讓 LLM 在無法回答問題時自動使用

**6. 工具清單 tools**

* 用於將上述兩個工具註冊為可由 LLM 使用的功能，型別為 function，以便配合 OpenAI 的 Function Calling 功能。

與 OpenAI 的互動邏輯及 **Gradio** 使用者介面設計：

**7. Chat 處理邏輯：函式 chat(message, history)**

**📌 功能說明：**

* 負責接收使用者訊息並與 OpenAI LLM 溝通，根據需要觸發工具（function calling）。
* 可處理多輪對話，並保留上下文記憶。

**🔧 處理流程：**

1. 建構 prompt：
   * 以一組系統訊息 system\_prompt 開頭（未在本檔案中定義，但預期用來定義 AI 的角色或語氣）。
   * 加入歷史對話紀錄 history
   * 加入目前使用者輸入 message
2. 呼叫 OpenAI Chat API：

*response = client.chat.completions.create(*

*model="gpt-4",*

*messages=messages,*

*tools=tools, # 註冊的工具列表*

*tool\_choice="auto" # 讓 LLM 自主決定是否觸發工具*

*)*

1. 檢查是否有工具被觸發（function\_call）：
   * 如果有，就依據回傳的 tool\_call 執行相對應的 Python 函式，如 record\_user\_details。
   * 並將執行結果再次送回 LLM，以完成回答。
2. 回傳 LLM 的自然語言回答文字（非 JSON）

**8. Gradio 介面：ChatInterface**

**📌 功能說明：**

* 使用 Gradio 的 ChatInterface() 提供一個 Web UI 聊天窗口。
* 對應函式為 chat(message, history)。
* 使用 type="messages" 以支援訊息物件格式，保留多輪對話紀錄。

**🖥️ 啟動方式：**

gr.ChatInterface(chat, type="messages").launch()

使用者可透過網頁輸入訊息，並與 LLM 模型互動，過程中如觸發特定需求會自動執行定義好的工具並記錄或發通知。

**✅ 總結功能邏輯圖（簡化版）**

使用者輸入 ➜ chat(message, history)

⤷ OpenAI GPT-4 (含 tools)

⤷ [自動判斷需觸發 function]

⤷ 執行 Python 工具 (record\_user\_details / record\_unknown\_question)

⤷ 回送執行結果給 GPT-4 ➜ 最終回應

⤷ Gradio Chat UI 顯示回答