

## 題目：個人電子郵件代理人 (AI Email & Scheduling Agent with MCP)

**背景：**你將為一位忙碌的經理開發一個具備自主決策能力的 AI Agent，協助其處理雜亂的收件匣，並透過標準化的數據協議管理會議預約。

**任務目標：**

1. 郵件分類與優先級：
  - 讀取 *emails.json*，將郵件分類為：急件、一般、詢價、會議邀約、垃圾。
  - 根據郵件上下文內容與寄件者身份，給予 1-5 分的優先級評等。
2. MCP Server 整合：
  - **核心要求：** 必須將 *calendar.json* 封裝為一個 MCP Server。
  - **工具定義：** MCP Server 必須提供至少 3 個 Tool：
    - `get_calendar_events`: 查詢行程。
    - `add_calendar_event`: 將確認後的會議寫入 JSON 檔案。
    - `delete_calendar_events`: 刪除行程。
  - **Agent 規範：** Agent 在判斷邀約無誤時，應主動調用 `add_calendar_event/` `delete_calendar_events` 完成排程自動化。
3. 智慧會議預約 (核心挑戰)：
  - 當偵測到「會議邀約」時，Agent 須透過 MCP 工具檢查行程衝突。
  - **多輪對話/時序推理：** 模擬場景 —— 若 Agent 先處理了郵件 A 並將其加入行程，隨後處理到時間重疊的郵件 B (即使郵件 B 也是重要邀約)，Agent 必須能偵測到「剛產生的衝突」，並對郵件 B 執行「婉拒」或「提議替代時間」。
  - **推理挑戰：** Agent 需結合「外部常識」。不論 *calendar.json* 是否已有行程，皆需能判斷週末與除夕等非工作時段，並給予適當的回覆建議 (如：主動提議改期)。
4. 安全與護欄 (Guardrails)：
  - 實作機制防止 Agent 在未經授權下做出合約/金錢承諾。
5. 其他說明
  - a. 模擬今日日期：2026 年 1 月 19 日

**交付物：**

- **原始碼 (透過 Github 提供)：** 包含 MCP Server 與 Agent 邏輯 (MCP 指定使用 python 的 fastMCP 套件) 以及環境配置文件 (ex: requirements.txt)。
- **執行結果：**
  - 提供每一封信的郵件分類、優先評級 (1~5)、信件回覆內容 (如果有回覆的話)
  - 更新後的 *calendar.json*
- **說明文件：**
  - 說明如何啟動程式 & MCP Server。
  - 演示 Agent 處理 13 封測試郵件的過程，並展現其在面對日期陷阱(國定假日) 時的決策流程。
  - 說明你如何處理「除夕與週末」的推理判斷邏輯。
  - 說明你如何設計 Prompt 或 Workflows 來避免模型幻覺。