**精簡版「碳智慧採購 MVP」──純本地執行 + 只用 OpenAI API**

一台筆電就能跑；離線元件全部開源（Tesseract、pymoo、Streamlit）。OpenAI API 只負責「理解文字 & 生成建議」。

**一、極簡資料流（5 個步驟）**

mathematica

複製編輯

PDF/EPD → OCR → 資料清洗 → 多目標優化 → OpenAI 智能加值 → 報告

| **#** | **功能** | **本地套件** | **呼叫 OpenAI?** |
| --- | --- | --- | --- |
| 1 | **OCR**：把 PDF 轉文字 | pdf2image → Tesseract | × |
| 2 | **抽取欄位**：供應商、材料、單價、數量、GWP | regex + pandas | ◯（可選：讓 GPT 整理雜訊表格） |
| 3 | **缺碳因子估算** | LightGBM 或簡單 rule | ✔（GPT 4o 推估 & 產信心分） |
| 4 | **多目標優化**（成本 × 碳 × 交期） | pymoo (NSGA‑II) | × |
| 5 | **商務話術 / 報告** | jinja2 產 Markdown→PDF | ✔（GPT 撰寫議價稿 & 摘要） |

**二、檔案／介面最小需求**

| **輸入** | **格式** | |
| --- | --- | --- |
| 報價 PDF × 多家 | 掃描或原生 PDF | |
| EPD PDF（若有） | 選填 | |
| 交貨期 .csv | item, supplier, eta\_day（人工輸入） | |
| **輸出** | | **說明** | |
| optimized\_plan.csv | | 成本、碳、交期 3 案比較 | |
| negotiation\_note.md | | GPT 生成的議價/替代料建議 | |
| summary\_report.pdf | | 一頁式執行摘要（pymoo 雷達圖 + GPT 結論） | |

**三、6 週單人排程（★ 必做）**

| **週** | **任務** | **關鍵套件 / API** |
| --- | --- | --- |
| 1 | ★ 建專案骨架 + Dockerfile + 範例資料 | poetry / conda |
| 2 | ★ OCR 管線：PDF → text → csv | pdf2image, pytesseract, pandas |
| 3 | ★ 欄位抽取 regex + 基本資料校驗 | pandas, pydantic |
| 4 | ★ pymoo 優化腳本；matplotlib 畫雷達 | pymoo, matplotlib |
| 5 | ★ GPT‑4o：  • 缺碳因子 prompt  • 議價稿 prompt | openai |
| 6 | ★ jinja2 報告 + Streamlit Demo △ CLI 參數/測試 | streamlit, jinja2, pytest |

**時程不夠？** 第 5 週只做「缺碳因子」；議價稿放文字樣板即可。

**四、OpenAI Prompt 範例（可直接改）**

python

複製編輯

system = """

你是一位綠色營建採購顧問，請根據材料名稱、製程關鍵字和供應地，預估其

Global Warming Potential (kg CO2e/t)。如果信心不足，用區間表示並給信心分(0~1)。

"""

user = """

材料: OPC 水泥

製程: 濕製程，煤佔 70%，天然氣佔 30%

產地: 台灣高雄

請用 JSON 回覆:

{ "gwp\_mean": ?, "gwp\_low": ?, "gwp\_high": ?, "confidence": ? }

"""

**五、本地環境一鍵啟動**

bash

複製編輯

git clone <repo>

cp .env.example .env # 填入 OPENAI\_API\_KEY

docker compose up --build

* localhost:8501 → Streamlit UI
* data/in/ 放 PDF；data/out/ 取報告