**交接文件說明**

## HW1

**主題**

葬送的芙莉蓮 — JSON 資料運用網頁

**成果內容**

* **首頁**：讀取 data.js，顯示作品資訊（標題、作者、簡介、預告片）。
* **角色列表**：讀取 characters.js，可依「陣營、種族、職業」篩選角色。
* **角色詳情**：依 URL id 顯示角色屬性、能力與描述。

**檔案**

* index.html（首頁）
* style.css（首頁樣式）
* main.js（首頁渲染與預告片切換）
* data.js（作品 JSON 資料）
* character.html / character.css（角色列表頁）
* characters.js（角色 JSON 資料）
* character-detail.html（角色詳情頁）

**使用方式**

* 以 VS Code Live Server 或其他靜態伺服器開啟 index.html。
* 首頁 → 角色列表 → 角色詳情，無需後端即可瀏覽。

## n8n自動化

**主題**

n8n 自動化工作流程（3 個）＋ 天氣查詢網頁

**成果內容**

* **weather\_fetch**
  + Webhook POST /webhook/send-weather
  + 呼叫 OpenWeather API → 回傳城市、國家、溫度、天氣、氣壓、雲層覆蓋率、座標
  + 可同時寄送「今日天氣」通知信件
  + 前端 weather.js 呼叫此流程顯示天氣與地圖
* **webhook\_test**
  + 測試 Webhook (POST /webhook\_test)
  + 回傳 { "body": { "demo": "OK" } } 用於確認連線正常
* **AI\_chat**
  + LangChain Agent 範例，模型為 **Ollama llama3.2:3b**
  + 用於展示 n8n 與 LLM 整合（目前未啟用）
* **weather 前端網頁**
  + index.html：城市下拉選單、自動定位查詢、結果＋Leaflet 地圖
  + style.css：網頁版面與 RWD 設計
  + weather.js：呼叫 n8n webhook、解析回傳、顯示天氣資訊與穿搭建議

**檔案**

* n8n workflows：weather\_fetch.json、webhook\_test.json、AI\_chat.json
* 前端程式：index.html、style.css、weather.js

**使用方式**

1. 啟動 n8n，確認 **weather\_fetch** 與 **webhook\_test** 為 Active。
2. 開啟 index.html（建議用靜態伺服器）。
3. 測試：
   * Webhook 測試 → POST /webhook\_test，回應 OK。
   * 天氣查詢 → POST /webhook/send-weather，回應天氣 JSON。
   * 網頁查詢 → 選擇城市或自動定位，即可顯示天氣與地圖。

## 新竹通

**主題**

新竹通後台退費流程與使用案例

**成果內容**

* **退費流程圖（泳道圖）**：描述退費申請、審核、執行與通知。
* **退費使用案例圖**：整理角色（使用者、支付單位、管理員）與功能。
* **取消退費流程圖**：顯示退費取消申請與結果回覆。
* **停車範例流程圖**：作為情境案例，幫助理解退費情境。

**檔案**

* 後台退費流程圖 (泳道圖).png
* 後台退費使用案例圖.png
* 後台取消退費流程圖 (泳道圖).png
* 新竹通\_停車範例.png

**使用方式**

* 圖檔為 PNG，可直接插入簡報或文件。
* 若需修改，請用 **draw.io** 開啟原始檔。

## 貿協後台—E化學院

**1. 頁面清單**

* 課程類別
* 課程（線上學堂）
* 課程（英文 e 點通）
* 課程章節
* 題庫音檔管理
* 八大類訓練
* 歷屆國內認證試題
* 英文章節訓練
* 海外心得
* 觀念交流

**2. 工項表（表格欄位）**

|  |  |
| --- | --- |
| **欄位名稱** | **說明** |
| 工項類別 | 功能所屬的分類（例：課程管理、內容管理…） |
| 工項名稱 | 該工項的名稱（例：課程類別管理） |
| 工項描述 | 工項的功能與作用簡述 |

**3. RA 文件（表格欄位）**

|  |  |
| --- | --- |
| **欄位名稱** | **說明** |
| 功能名稱 | 該功能的名稱 |
| 功能項次 | 項目編號 |
| 使用權限 | 哪些角色可使用 |
| 雛型系統 | 是否有參考或原型 |
| 程式編碼 | 系統程式代號（SA/SD 填寫） |
| 使用者情境 | 使用者如何操作該功能的情境 |
| 功能目的 | 功能設計的目標與價值 |
| 作業說明 | 操作流程與規格說明 |
| 資料表 | 關聯的資料表結構 |
| 畫面說明 | UI 頁面配置與說明 |

**4. 檔案**

📂 公司雲端連結：[改造計畫](https://drive.google.com/drive/folders/1_7foCIPZAG-eT1BA_A0oFgqfG-Mi_v7G?usp=drive_link)

## 工時系統

**1. 資料夾結構**

工時系統專案文件與程式分為四個子資料夾：

1. AI工項
2. RA文件
3. API
4. 資料庫

**2. AI工項**

檔案

* 任務管理模組.docx
* 任務視圖切換模組.docx
* 專案管理模組.docx
* 彈窗操作模組.docx

說明

* 格式：工項表（工項類別、工項名稱、工項描述）。
* 內容：初版功能規劃，包含專案管理、任務管理、任務檢視切換、彈窗操作等模組。

**3. RA文件**

檔案

* 任務管理模組.docx
* 任務視圖切換模組.docx
* 專案管理模組.docx
* 彈窗操作模組.docx

說明

* 格式：RA 文件（功能名稱、功能項次、使用權限、雛型系統、程式編碼、使用者情境、功能目的、作業說明、資料表、畫面說明）。
* 內容：各模組的需求分析與規格，輔助後續系統設計與前端實作。

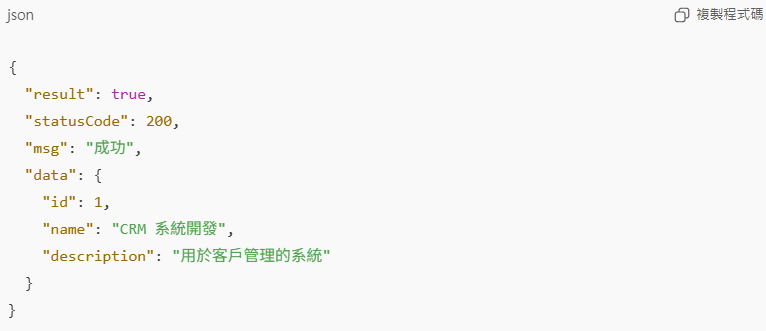
**4. API**

檔案

* API修改版.xlsx
* TimesheetSystem

說明

* 架構：
  + Controller：處理 HTTP 請求，呼叫 Repository 與資料庫。
  + Dto：
    - Request Dto → 定義請求欄位格式。
    - Response Dto → 分為兩層：
      * BaseResponse：統一回傳格式（result、statusCode、msg）。
      * DataResponse：data 內容（例如專案清單、工時紀錄）。
* 涵蓋範圍：
  + 專案管理 (TopTasks)：新增、更新、刪除、列表、詳情
  + 子任務管理 (SubTasks)：建立、更新、刪除、樹狀/子層查詢、詳情
  + 工時紀錄 (WorkHours)：新增、更新、刪除、查詢
  + 員工管理 (Employees)：查詢列表
* Request/Response 格式：
  + Request：統一採 POST，Body 使用 JSON 格式。
  + Response（範例）：



**5. 資料庫**

檔案

* 工時系統\_資料庫建立程式碼.sql
* 實際工時索引.sql
* 工時系統\_員工測試資料.sql
* 工時系統\_專案測試資料.sql
* 工時系統\_子任務測試資料.sql
* 工時系統\_工時測試資料.sql
* 測試資料.sql
* 資料庫.xlsx

說明

* Schema：包含四大核心資料表
  + Employees（員工）
  + TopTasks（專案）
  + SubTasks（子任務，支援階層結構）
  + WorkHours（工時紀錄）
* 索引：SubTasks(TopTask\_Id)、WorkHours(SubTask\_Id)
* 測試資料：已提供員工、專案、子任務、工時多組測試數據
* 使用方式：
  + 在 SQL Server 執行 工時系統\_資料庫建立程式碼.sql 建立資料庫
  + 執行 實際工時索引.sql 建立索引
  + 匯入測試資料（可選）
* .NET 連線字串 (DB First)：

"ConnectionStrings": {

"WebDatabase": "Server=127.0.0.1;Database=TimesheetSystem\_new;User ID=你的帳號;Password=你的密碼;TrustServerCertificate=true"

}

⚠️ 帳號/密碼請依照個人環境修改。

**6. 備註**

* 本文件為 初版交接，後續規格與程式碼可能有修改，請以實際開發版本為準。
* 程式已開放區網存取，CORS 設定允許 http://localhost:8000，前端可在本機開發並測試 API。
* 資料庫可透過附帶的 SQL 腳本自行建立，帳號密碼需依環境調整。

## 報表轉換

**主題**

HTML 報表轉換工具（支援 PDF 與 Excel 匯出）

**成果內容**

* 特色
  + 市面上許多工具無法正確處理 colspan / rowspan，本工具可完整支援。
  + 使用者可直接把整段 HTML 程式碼貼上，系統會自動偵測 <table> 區塊並轉換。
  + PDF：所有表格合併在同一份文件。
  + Excel：每個表格分成獨立工作表（表格1、表格2…）。
* 檔案
  + index.html → 原始版本，僅支援 PDF 匯出。
  + index2.html → 進階版，新增 Excel 匯出。
  + 測試頁：test1.html、test2.html（單表格）、test3.html（雙表格）。
  + settings.json → Live Server 設定（port 5501）。

**使用方式**

1. 開啟 index.html 或 index2.html。
2. 將含有 <table> 的 HTML 程式碼貼入輸入框。
3. 點「預覽 HTML」檢查格式是否正確。
4. 點「下載 PDF」或「下載 Excel」完成匯出。