

Subscribe to DeepL Pro to edit this document.  
Visit [www.DeepL.com/pro](https://www.deepl.com/pro?cta=edit-document) for more information.



**模組化課程**

**課程作業評量規範**

**（草案）**

**模組詳情**

|  |  |  |
| --- | --- | --- |
| **模組代碼** UFCE3Q-30-3 | **開課 主修（2025/26學年第一學期）**  **全日制班級** | **模組名稱**  進階網頁開發 |
| **模組負責人** | | **模組導師 劉思明** |
| **單元與元素編號**  個人作品集 | | **權重：** 100% |
| **元素描述** 建構一個能展示在實際情境中應用進階網頁開發技術能力的網路應用程式。 | | **總作業時間** |

**日期**

|  |  |
| --- | --- |
| **發布給學生日期**  2025年9月7日 | **交回學生日期** |
| **提交地點**  電子檔 ：Moodle | **草稿提交日期與時間：**  2025年11月2日（星期日）23:55 |
| **正式提交日期與時間：**  2025年12月7日（星期日）23:55 |

**交付成果**

|  |
| --- |
| 符合規範之運作應用程式與文件。 |

**模組負責人簽署**

|  |
| --- |
|  |

**課程概述**

本**個人**作業要求學生設計、建置並部署網頁應用程式，展現其在實際情境中運用進階網頁開發技術的能力。作為作品集組成部分，需提交文件對應用程式的設計、實作、效能及遵循網頁標準與最佳實踐的情況進行批判性評估。透過此個人作業下列將評估模組學習成果：

* **MO1** 分析與評估網頁標準、通訊協定及新興技術，展現於網頁應用程式開發中運用物件導向程式設計技術之能力。
* **MO2** 識別並應用常見軟體模式與網頁架構於實務中。
* **MO3** 於整個網頁開發專案生命週期中，展現運用當代工具、技術及網頁框架的熟練度。
* **MO4** 運用高效開發方法、測試策略與軟體文件化實務，創建並批判性檢視網頁應用程式，展現對開發流程中這些實務重要性的理解。

**需完成之任務**

您需完成以下任務：

1. 建置提供 RESTful 網路服務的伺服器，用於搜尋及維護由 data.gov.hk 提供的「流動郵政局」資料，
2. 建置基於網頁的應用程式作為（1）中 RESTful 網路服務的客戶端。該客戶端提供資訊查詢、視覺化呈現及維護功能。
3. 執行、記錄並分析一系列測試，以客觀評估整個應用程式堆疊（包含資料儲存、網路服務及客戶端應用程式）的適用性、可用性與穩健性，
4. 編寫詳盡文件，涵蓋應用程式架構、採用工具與平台、提供的網路服務及客戶端功能、設計開發評估、後續優化建議，以及您對整個專案週期的反思；
5. 進行簡報與實機演示，展示您的工作成果與應用產品。

本課程作業包含**五個**部分，詳見後續章節說明。

此為**個人**作業。

**第一部分：資料轉換（3%）**

為建立行動郵局資訊資料庫，請透過以下連結下載JSON格式資料集：

<https://data.gov.hk/en-data/dataset/hk-hkpo-hkpo_ds01-hkpo-mobile-office>

建議參閱上述連結提供的資料字典，以理解資料欄位與JSON結構。以下展示資料集的前兩筆記錄：

**{**

**"data":[**

**{**

**"mobileCode":"1",**

**"locationTC":"深井",**

**"locationSC":"深井",**

**"地址繁體中文":"青山公路深井段，海韻花園外。",**

**"nameSC":"流動郵政局 1",**

**"districtSC":"荃灣區",**

**"地址SC":"青山公路深井段，海韻花園外。",**

**"closeHour":"09:30",**

**"nameTC":"流動郵政局 1",**

**"districtTC":"荃灣區",**

**"latitude":"22.36774",**

**"openHour":"09:00",**

**"星期代碼":1,**

**"nameEN":"流動郵政局 1",**

**"districtEN":"荃灣",**

**"locationEN":"深井",**

**"地址EN":"青山公路深井段（鄰近萊茵花園）",**

**"序號":1,**

**"longitude":"114.06233"**

**},**

**{**

**"mobileCode":"1",**

**"locationTC":"黃金海岸",**

**"locationSC":"黃金海岸",**

**"地址繁體中文":"屯門黃金海岸，黃金海岸商場外。",**

**"nameSC":"流動郵政局 1",**

**"districtSC":"屯門區",**

**"地址SC":"屯門黃金海岸，黃金海岸商場外。",**

**"closeHour":"10:05",**

**"nameTC":"流動郵政局 1",**

**"districtTC":"屯門區",**

**"latitude":"22.37184",**

**"openHour":"09:50",**

**"星期代碼":1,**

**"nameEN":"流動郵政局 1",**

**"districtEN":"屯門",**

**"locationEN":"香港黃金海岸",**

**"地址EN":"屯門黃金海岸黃金廣場外",**

**"序號":2,**

**"longitude":"113.99305"**

**},**

**…**

**]**

**}**

您需開發伺服器端程式，從上述連結取得資料集並將資料遷移至資料庫。可先下載資料集，再將資料遷移至資料庫；或採用單一步驟即時下載資料並同步遷移至資料庫。

**注意：**

* 此資料集包含繁體中文、簡體中文及英文內容。您的網頁應用程式必須支援英文，但可選擇性支援繁體中文與簡體中文（詳見第 3.3 節）。
* 請謹慎設計資料庫架構，確保其與資料集相符並充分支援伺服器 API。**建議為每筆行動郵局紀錄建立唯一識別碼，並運用該識別碼執行 CRUD 操作時定位目標紀錄。**

**第二部分：RESTful網路服務設計與實作（總分佔22%）**

須開發 RESTful 網路服務，用於檢索與維護第一部分建置之資料庫中的行動郵局紀錄。伺服器傳送至客戶端的資料必須採用 JSON 格式，且 JSON 回傳內容不得包含任何客戶端格式化指令（如 HTML 標籤與 CSS 規則）。

網際網路服務之實作須採用PHP或NodeJS，不得使用CodeIgniter、Laravel等高階框架。行動郵局紀錄儲存可自行選擇採用本地端MySQL資料庫或雲端資料庫平台。

網路服務必須採用以下HTTP方法：

|  |  |  |
| --- | --- | --- |
| HTTP 方法 | 用途 | 標記 |
| POST | **建立**新記錄。 | 3% |
| GET | **檢索**記錄詳情，例如地區、地址及營業時間。 | 9% |
| PUT | **更新**記錄資訊。 | 4% |
| 刪除 | **刪除**現有記錄。 | 3% |

**2.1 應支援的 API 數量**

雖然強烈建議您建立全面的 API 以支援各種客戶端請求，但您亦可參照下表，該表列出伺服器針對不同操作類型必須提供的 API 最低數量與要求。

|  |  |  |
| --- | --- | --- |
| API 類型 | 最低 API 數量 | 備註 |
| 建立 | 1 | * 建立行動郵局記錄。 * 資料值必須透過 HTTP 請求本體傳送。 |
| 檢索 | 3 | * 允許客戶端使用 不同條件搜尋記錄，例如透過記錄ID進行檢索。   + 使用記錄ID檢索、   + 搜尋特定行政區內的行動郵局，   + 等等。 * 必須提供支援多重搜尋條件的API，例如查詢荃灣區於14:00開放的流動郵政局。 |
| 更新 | 1 | * API 必須支援記錄的局部更新——僅需將需更新欄位的數據值傳送至伺服器。換言之，伺服器不應「覆寫」整筆記錄，而僅更新待修改的個別欄位。 |
| 刪除 | 1 | * 刪除行動郵局記錄。 * 指定待刪除記錄的識別碼可透過 URL 或 HTTP 請求本體傳送。 |

**2.2 關於服務調用的補充說明**

您必須精確定義 API（URL 格式）以呼叫網路服務。

* + 若使用 GET 方法進行檢索，可能格式如下：

http://www.myserver.com/mobilepost?district=Yuen+Long

上述網址要求伺服器回傳元朗區內流動郵政局的清單。

* + 在 GET 與 DELETE 操作中，參數值應附加於 URL 還是透過 HTTP 本體傳送，可由您自行決定。
  + PUT與POST操作時，參數值必須透過HTTP主體傳送。

**注意：**

* + 上述 API 僅為範例。您可依據所需操作與參數自訂 API 格式。
  + API 必須在報告中精確定義並完整記錄（參見第 5 部分），否則客戶端應用程式開發者將無法呼叫相關服務。
  + 為簡化作業流程，您的網路服務僅需支援英文。但應用程式可將繁體中文與簡體中文列為進階功能（參見第3.3節）。

**2.3 資料格式補充說明**

伺服器傳回客戶端的任何資料皆須採用 JSON 格式。以下為客戶端查詢元朗區行動郵局清單時，伺服器可能傳回的 JSON 範例。此格式僅供參考，您可自行定義 JSON 格式。

{

"header":{

"success":true,

"訊息":"檢索到12筆記錄",

"錯誤代碼":"0000",

"錯誤訊息":"未發現錯誤"

},

"結果":[

{

"mobileCode":"1",

"locationTC":"洪水橋",

"locationSC":"洪水橋",

"地址繁體中文":"元朗洪水橋大路與青山公路洪水橋段交界，翠珊園附近。",

"nameSC":"流動郵政局 1",

"districtSC":"元朗區",

"地址SC":"元朗洪水橋大路與青山公路洪水橋段交界，翠珊園附近。",

"closeHour":"14:40",

"nameTC":"流動郵政局 1",

"districtTC":"元朗區",

"latitude":"22.43353",

"openHour":"14:10",

"星期代碼":1,

"nameEN":"流動郵政局 1",

"districtEN":"元朗",

"locationEN":"洪水橋",

"地址EN":"鄰近翠山園，位於元朗洪水橋主幹道與青山公路交界處（洪水橋）",

"序號":7,

"longitude":"113.99781"

},

{

"mobileCode":"1",

"locationTC":"洪水橋",

"locationSC":"洪水橋",

"addressTC":"元朗洪水橋大路與青山公路洪水橋段交界，翠珊園附近。",

"nameSC":"流動郵政局 1",

"districtSC":"元朗區",

"地址SC":"元朗洪水橋大路與青山公路洪水橋段交界，翠珊園附近。",

"closeHour":"15:25",

"nameTC":"流動郵政局 1",

"districtTC":"元朗區",

"latitude":"22.43353",

"openHour":"14:55",

"星期代碼":3,

"nameEN":"流動郵政局 1",

"districtEN":"元朗",

"locationEN":"洪水橋",

"地址EN":"鄰近翠山園，位於元朗洪水橋主幹道與青山公路交界處（洪水橋）",

"序號":7,

"longitude":"113.99781"

},

……

]

}

**2.4 錯誤處理 (3%)**

網路服務必須以優雅的方式處理錯誤。當 API 被錯誤呼叫時（例如缺少必要參數或參數格式錯誤），伺服器必須向客戶端返回相關且具意義的錯誤代碼與訊息。同樣地，錯誤代碼與訊息採用 JSON 格式呈現。以下為檢索服務參數無效時的範例：

http://www.myserver.com/mobilepost/district=

|  |
| --- |
| {  "header":{  "success":false,  "err\_code":"0120",  "錯誤訊息":"缺少區值"  }  } |

您需根據自身網路服務定義專屬的錯誤代碼與訊息。錯誤代碼與訊息必須精確定義並於報告中完整記錄（參見第五部分）。上述 JSON 回傳僅為範例，您可自行定義 JSON 格式

**第三部分：客戶端應用程式設計與實作（總分佔30%）**

**3.1 行動郵局紀錄檢索與維護網頁應用程式**

設計並實作客戶端網頁應用程式，運用第一、二部分伺服器提供的 API 進行行動郵局紀錄的檢索與維護。客戶端應用程式必須採用 **Angular** 框架作為開發平台，嚴禁使用 React、NextJS、VUE 等其他框架開發，違者將導致整份作業成績記零分。Angular客戶端應用程式除自動生成的app元件外，必須至少包含四個Angular元件。

**注意：**

* + 允許使用 Semantic UI、Bootstrap 等 UI 函式庫，但僅限於提供視覺效果之用途。
  + 建議使用Materials UI但非強制要求。
  + jQuery僅限用於UI效果呈現。嚴禁使用jQuery執行表單驗證及向伺服器API發送AJAX請求。

**3.2 使用者介面與體驗**

* + 您可自由選擇使用 UI 函式庫（如 Semantic、Bootstrap 或 Material），但嚴禁將此類函式庫用於 UI 外觀以外之功能。例如表單驗證必須透過 Angular 反應式表單模組內建機制完成，而非依賴 UI 函式庫。嚴格限制客戶端頁面的刷新與重新載入。換言之，您需透過妥善的設計與專案架構，開發符合 Angular 應用程式本質的單頁應用程式（SPA）。
  + 您可設計專屬的用戶介面風格，但介面必須具備易用性，並在字型尺寸、樣式、色彩運用等方面保持視覺一致性。網頁呈現應簡潔俐落且操作直覺。
  + 您必須非常謹慎地設計操作流程。例如，典型的操作流程是搜尋特定區域內的行動郵局，然後從列出的記錄中選擇一項查看詳細資訊。極差的設計是要求使用者記住ID，再前往另一頁面重新輸入ID才能查看詳細資訊。
  + 行動郵局的維護頁面可與檢索頁面分離，因兩者本質上屬於不同功能。但為確保更佳的使用者體驗，仍應避免過度刷新與重新載入。

**3.3 進階功能設計**

* + 鼓勵您設計並實作進階功能，例如Google地圖與路線規劃、多語言支援、響應式網頁等。您可將此類功能記錄於文件中，並於演示時加以說明。

**第四部分：測試（10%）**

需設計並執行多項測試，驗證網頁應用程式是否以適當品質水準達成預期功能。RESTful網路服務與客戶端應用程式均須納入測試範圍。請提交**獨立測試報告**詳述執行內容，包含：

* + 測試策略、測試計畫、測試案例、預期結果、
  + 實際結果及其分析、結果解讀與測試洞察。

測試計畫至少涵蓋兩大面向：(1) 正確性與穩健性，(2) 使用者驗收（即需執行使用者驗收測試，UAT）。進行UAT時，須列出受測者組合清單，並說明其作為目標使用者的有效性與代表性依據。

可考慮將原始測試結果置於附錄，避免佔用主報告過多篇幅。

**第五部分：成果展示、實作演示、文件記錄與反思（35%）**

**簡報：**

* + 須進行5分鐘簡報，內容涵蓋專案背景與目標、已完成事項（如架構設計、伺服器API設計、客戶端應用程式設計、採用工具技術、關鍵評估、後續優化等）。
  + 評分依據將涵蓋網頁應用程式建置成果，最關鍵在於您對整個開發流程的批判性反思與評估。
  + 請勿使用大量螢幕截圖描述客戶端應用程式功能，此類內容在演示環節已充分呈現。請聚焦於架構設計、設計理念、評估與反思等高階議題。
  + 儘管演示技巧與簡報視覺品質不列入評分項目，但作為學位課程學生，應具備高水準的 品質與充分準備的簡報。特別嚴禁任何形式的照稿念稿行為，無論使用紙本或數位裝置。

**實作展示：**

* + 需進行25分鐘應用程式演示，內容包含：(1)伺服器端資料轉換程式、(2)伺服器API介面、(3)客戶端應用程式。
  + 數據轉換環節需從空數據庫開始，展示程式如何將數據遷移至數據庫。轉換前後的數據庫內容均需呈現。
  + 針對伺服器 API，需使用 **Postman 或類似工具**向伺服器發送 CRUD 操作請求，並直接展示伺服器返回的 JSON 內容，包含正常資料與錯誤代碼/訊息。
  + 請注意，整個示範將由學生主導，以展示功能與特色，而非由評分者引導。
  + 請預先規劃演示步驟，並著重說明特殊功能與設計特色。
  + 評分依據包含：API完整性與相關錯誤處理機制、客戶端應用程式功能性與易用性，以及透過問答環節展現的開發流程反思能力。

**問答環節：**

* + 簡報與演示結束後，將進行10分鐘問答環節。評審將提出問題以釐清您的工作與解決方案，同時提供機會讓您進一步闡述參與專案過程中的思考與反思。
  + 請注意，問答環節對展現批判性思維至關重要，將對本次評分佔比產生顯著影響。

簡報、實作展示及問答環節的具體日期與時間將另行公告，暫定於2025年12月7日當週舉行。

請務必準時加入會議並備妥所有設備。評審不會等待您設置或載入簡報與展示材料，您必須嚴格遵守時間限制。若未能於規定時限內完成所有內容，恕不給予額外補救時間；同樣地，遲到將導致自身時間損失且不予補救。

**專案報告：**

* + 須提交書面報告詳述伺服器API與客戶端應用程式的設計及技術細節。請審慎思考報告應涵蓋之重要核心內容，並以邏輯架構組織內容。必要時可將內容劃分為多個章節與子節。務必仔細詳閱本課程作業規範及評分標準。
  + 報告結構與內容雖由您自行決定，但下列項目必須包含在內。（請注意，此清單僅列出您 必須納入報告的內容，並非強制規定的報告結構。您可根據自身寫作需求設計報告架構。）
  + **封面頁** – 必備項目，需標示模組代碼與模組名稱、小組成員姓名及學號等資訊。
  + **目錄** – 建議添加章節與小節編號以清晰呈現報告架構。目錄中每項條目均須標註對應頁碼。
  + **圖表目錄** – 若報告含大量圖表，可考慮增列圖表目錄。每幅圖表需編號並附說明文字，圖表說明置於圖表底部。圖表目錄應列出各圖表編號、說明文字及對應頁碼。
  + **表格目錄** – 與圖表目錄類似，但適用於表格。請注意表格說明文字應置於圖表上方而非下方。
  + **背景說明** – 問題情境為何？應用程式的預期目的為何？本專案已完成哪些工作？
  + **採用的軟體模式與網頁應用程式架構**
    - 能否列舉並描述不同網頁應用架構？
    - 您在工作中採用了何種架構？其採用原因為何？
    - 能否列舉並說明您工作中採用的軟體模式？
    - 架構元件如何協同運作以達成網頁應用的最終目標？
  + **採用的開發工具、函式庫與框架**
    - 您採用了哪些開發工具、函式庫與框架？
    - 這些工具、函式庫與框架具備哪些特性？
    - 為何選擇這些工具應用於您的專案？
    - 相較於使用低階程式設計結構（如純 JavaScript 結合 HTML/CSS）作為獨立且分離的程式設計單元來開發網頁應用程式，採用這些工具、函式庫與框架有何優勢？
  + **RESTful API** 規範，包含請求與回應格式及錯誤代碼規範。若需定義大量 API，可考慮將完整 API 規範列為附錄，並在核心報告內容中提供簡要摘要。
  + **客戶端**應用程式所提供之客戶端**功能規範**。
  + **產品**評估——從多角度批判性思考網頁應用的優劣勢，包括但不限於：伺服器 API 的完整性、客戶端應用功能與易用性、網頁架構的利弊、所選開發工具與平台的優缺點，以及透過本專案所獲取的網頁應用開發經驗等。   
      
    **提醒**：評估要求您針對 專案成果進行自主批判性思考與反思，而非僅陳述事實性內容（如Angular框架定義、RESTful架構原理等）。您需分享透過本專案實踐所獲得的經驗與教訓。若僅提供事實性內容，將導致評分極低甚至不予計分。
  + **針對未來改進的建議**，應涵蓋伺服器端與客戶端應用程式，並可包含整體應用程式架構的優化方案。每項建議需附上合理依據，說明該建議值得在未來實施的理由。   
      
    **提醒**：未來改進建議須基於問題情境且與工作相關，而非泛泛而談。不良範例如：(1)僅提出「提升使用者友善度」卻未說明具體方式，(2)主張「強化安全性」卻未闡明方法與理由等。
  + 結論——必備要素。引用Grammarly.com定義：*「將分析要點精煉為簡潔總結*」且*「有效的結論段落應明確向讀者傳達*：*您已完成預設的論證目標*。」[來源：grammarly.com]
  + **參考文獻與文內**引用——您不可能完全憑空創作。必然引用過他人軟體平台、概念、論述等資源。因此必須標註參考文獻與文內引用，讓讀者知悉資訊來源，並可自行深入探索。引用格式須**採用UWE-哈佛引用風格，**詳見下方連結。

[**https://www.uwe.ac.uk/study/study-support/study-skills/referencing/uwe-bristol-harvard**](https://www.uwe.ac.uk/study/study-support/study-skills/referencing/uwe-bristol-harvard)

* + 報告補充說明：
  + **禁止使用螢幕截圖**——您將進行實機演示，本報告並非操作指南。因此應用程式截圖不應作為核心內容呈現，若確有需要，請另行收錄於附錄。
  + 無需**列出**原始碼——無論作為核心內容或附錄，皆無需提供原始碼。若需使用程式碼輔助說明內容，請嚴格控制數量，並確保程式碼僅用於闡釋高階概念。

**草稿提交**

為確保研究進度與方向正確，請**於2025年11月2日23:55前將**報告草稿及原型演示影片提交至Moodle平台。

請注意草稿提交屬強制要求，此舉將有助您確認研究方向正確性，並在最終提交前完善成果。過往經驗顯示，未提交草稿的學生最終產出內容往往不符要求，因而獲得極低評分。若選擇不提交草稿，後果需自行承擔。

**提交規範**

須提交下列五項內容：

(1) 簡報PPT

(2) 伺服器端網路服務 - 原始碼及輔助檔案

* 請提交所有原始檔案，包含建立與操作資料庫的SQL指令，以及建置與部署網路服務所需之完整檔案。若使用外部函式庫，亦須一併提交。

(3) 客戶端應用程式 – 原始碼與輔助檔案

* 提交所有原始檔案，包含 Angular 專案及建置與部署客戶端應用程式所需之所有檔案。若使用任何外部函式庫，亦須一併提交。

針對(2)與(3)，請將所有檔案與資料夾壓縮為單一ZIP檔案（僅限zip格式，禁止使用rar或7zip）。提交之原始碼**必須**與演示時使用的版本**完全一致**。

(4) 測試報告

* 僅接受 MS-Word 或 PDF 格式。

(5) 專案報告

* 僅接受 MS-Word 或 PDF 格式。

針對第(4)及(5)項，若使用任何線上編輯工具，請確保匯出檔案的格式與行距均正確無誤。

**重要提示：**學生有責任確保提交的ZIP及其他檔案能成功解壓縮並開啟。若提交檔案無法正常開啟，相應評核任務將被判為零分**。**

**評分標準**

|  |  |
| --- | --- |
| **第一部分：數據轉換（3%）** | |
| 0 | * 未能下載資料來源並遷移至MySQL資料庫或任何雲端資料庫平台。 |
| 1 | * 能夠下載資料來源並遷移至 MySQL 資料庫或任何雲端資料庫平台。 * 流程存在部分缺陷，例如非即時處理，需透過獨立指令步驟介入。 * 資料庫架構可能無法完整涵蓋資料來源的所有欄位。 |
| 2 | * 透過單一指令即時成功將資料遷移至 MySQL 資料庫或任何雲端資料庫平台。 * 完善的資料庫設計可完整支援資料來源的所有欄位。 |
| 3 | * 卓越的實作品質，可能包含進階功能，例如匯入資料項目的報告或異常狀況警示。 |

|  |  |  |
| --- | --- | --- |
| **第二部分：RESTful 網路服務之設計與實作（總分佔比 22%）** | | |
| 檢索  (9%) | 0-2 | * 功能部分可用，但檢索記錄資訊時出現錯誤。 |
| 3-4 | * 功能正常，但僅支援極有限或完全不支援檢索條件。 |
| 5-7 | * 支援多種檢索條件 * 合理的 API 格式 * 支援部分錯誤檢查 * 以 JSON 格式回覆客戶端應用程式 |
| 8-9 | * 複雜搜尋條件 * 良好的 API 格式 |
| 更新  (4%) | 0-1 | * 更新記錄時部分功能正常但存在錯誤。 * 未使用 PUT 方法者不予計分，此部分其他評分標準概不適用 |
| 2-3 | * 使用 PUT 方法 * API格式合理 * 支援部分錯誤檢查 |
| 4 | * 良好的 API 格式 * 僅將需更新的欄位傳送至伺服器 |
| 新增  (3%) | 0-1 | * 新增記錄功能部分失效，存在錯誤。 * 非POST提交 – 若未使用POST方法，則此部分不予計分，其他評分標準概不適用。 |
| 2-3 | * 使用 POST 方法 * API格式合理 * 支援部分錯誤檢查 |
| 移除  (3%) | 0-1 | * 刪除記錄時部分功能正常但存在錯誤。 * 未刪除 – 若未使用 DELETE 方法，則給予 0 分，此部分其他評分標準不予考量。 |
| 2-3 | * 使用 DELETE 方法 * API格式合理 * 支援部分錯誤檢查 |
| 錯誤處理  (3%) | 0-1 | * 無或僅有極有限的錯誤代碼與訊息 |
| 2-3 | * 具備完善且複雜的錯誤代碼與訊息 * 錯誤代碼設計完善（例如：依錯誤類型採用階層式編號） * 清晰且具實質意義的錯誤訊息，能明確描述錯誤性質與成因 |

|  |  |  |
| --- | --- | --- |
| **第三部分：客戶端 Angular 應用程式開發（總分佔 30%）** | | |
| 記錄資訊檢索  (10%) | 0-2 | * 部分功能正常，但檢索記錄時會發生錯誤。 |
| 3-5 | * 功能正常，具備合理介面供指定搜尋條件 * 非 SPA – 若非 SPA 項目，此部分最高可得 5 分，不受其他評分標準影響。 |
| 6-8 | * SPA * 具備良好介面，能讓使用者輕鬆精確地設定搜尋條件 * 當伺服器回傳錯誤時，能向使用者提供合理的反饋 |
| 9-10 | * SPA * 複雜的搜尋條件設定 * 清晰直觀的記錄詳情格式化，可搭配進階功能如運用JQuery強化視覺效果。 |
| 記錄維護  (8%) | 0-2 | * 部分功能可用但存在錯誤 |
| 3-6 | * 功能正常且具備合理使用者介面 * 具備基礎錯誤檢查機制，並在發生錯誤時向使用者提供反饋 |
| 7-8 | * 維護作業成功完成後提供充分的用戶回應 * 完善的錯誤檢查機制，並於發生錯誤時向使用者提供反饋 |
| 可用性與使用者體驗 （5%） | 0-1 | * 工作流程設計欠佳 * 操作流程對使用者而言不夠直觀 |
| 2-3 | * 良好的介面與工作流程設計 * 使用者能直觀且輕鬆執行所需操作。 |
| 4-5 | * 投入大量心力於介面與流程設計，造就極佳的可用性與使用者體驗。 * 設計風格統一且時尚，可能包含部分進階功能。 |
| 額外且具挑戰性的功能  (7%) | * 依據實用性與實作難度評分 * 每項功能通常應獲2分，最高3分 * 範例：多語言支援、地圖與路徑導航、記錄分頁與排序、響應式網頁設計 | |

|  |  |  |
| --- | --- | --- |
| **第四部分：測試（總計10%）** | | |
| 測試步驟與執行  (5%) | 0-1 | * 僅執行少量基礎測試案例，未提供解讀甚至原始結果。 |
| 2-3 | * 執行若干測試案例，但未達全面性要求。 |
| 4-5 | * 執行相當全面的測試，涵蓋不同面向。 |
| 測試文件與分析  (5%) | 0-1 | * 對結果的分析與詮釋不足且薄弱。 |
| 2-3 | * 詳細記錄原始結果，並包含部分結果分析與詮釋。 |
| 4-5 | * 透過全面分析與詮釋，深入洞察產品品質、弱點及所需改進之處。 |

|  |  |  |  |
| --- | --- | --- | --- |
| **第五部分：簡報、實作展示、文件記錄與反思（總計35%）**  \*簡報與實作演示旨在驗證學生對模組學習成果的掌握程度。評分依據為內容本身，而非簡報與演示技巧。 | | | |
| 軟體模式與架構（5%） | 0-1 | | * 僅提供極有限或完全缺乏各類網頁架構資訊。 * 無法展示自身作品的網頁架構。 |
| 2-3 | | * 能夠識別並比較不同網頁架構。 * 清晰記錄網頁應用程式的架構，明確標示各元件及其 互動關係。 |
| 4-5 | | * 深入探討所採用架構的優缺點，可能還提及先進網頁應用程式架構的現況與未來趨勢。 |
| 開發工具、函式庫與框架  (5%) | 0-1 | | * 僅陳述一些瑣碎工具（例如整合開發環境），卻未說明它們在專案中的作用。 |
| 2-3 | | * 能夠辨識專案中採用的高階工具、函式庫與框架，並進行簡要說明。 |
| 4-5 | | * 對工具、函式庫及框架進行全面性探討，特別著重其如何促進開發流程、實現網路架構，並確保未來維護性。 |
| 網路服務 API 規範  (5%) | 0-1 | | * API 格式說明不足。 * API 設計欠佳，可能無法支援必要參數 |
| 2-3 | | * API設計良好，格式規範清晰明確 * 為 API 使用者提供充分細節 |
| 4-5 | | * 完整API規格，涵蓋所有必要細節。 * 具備良好可讀性，讓讀者能輕鬆掌握 API 功能 |
| 功能規範說明  (5%) | 0-1 | | * 僅提供功能的平面清單，缺乏詳細說明或合理依據 |
| 2-3 | | * 具備充分說明與依據的分類功能清單 * 功能清單可能不夠全面，僅提供較粗略的功能項目 |
| 4-5 | | * 具備適當細分層級的完整功能清單，每項功能皆附有恰當說明與依據。 |
| 評估  (5%) | 0-1 | | * 僅列舉事實要點，未觸及開發工作或產品本身。 |
| 2 | | * 基於實際經驗與批判性評估提出若干相關觀點，但尚不全面。 |
| 3-4 | | * 基於本專案經驗與教訓的良好探討。 * 內容聚焦本專案經驗，而非泛泛而談。 |
| 5 | | * 卓越的討論展現學生對整個開發週期具備深度批判性思考，並能在專案執行後進行反思。 * 內容聚焦於專案實務，而非泛泛而談。 |
| 未來改進方向之探討  (5%) | 0-2 | | * 僅提出若干瑣碎觀點，缺乏具體說明。 |
| 3-4 | | * 提出相當多具實質意義的改進建議，並附有充分說明以闡明這些改進值得考慮的價值。 |
| 5 | | * 涵蓋多面向的全面性建議，附有詳盡說明與論證依據。 |
| 格式與整體報告品質  (5%) | 0-2 | * 報告內容不完整或僅提供極少細節。 * 參考文獻與文內引用缺失或數量極少。 | |
| 3-4 | * 細節程度適中，但組織與格式尚可 * 提供部分參考文獻與文內引用，但未達全面性要求 | |
| 5 | * 組織與格式極佳，細節充分 * 大量參考文獻與文內引用。 | |

**END. (草稿)**