**Coding101 > 2025**

**競賽時程**

* 報名資訊：2024/9/30 (一) 起至2025/1/12 (日) 23:59
* 初賽交件：2024/12/2 (一) 至2025/1/12 (日) 23:59前繳交應繳文件
* 初賽公布：預定於2025/2/10 (一) 公告初賽結果並告知複賽時程及比賽方法
* 複賽交件：詳細資訊請參閱複賽公告
* 複賽比賽：預定於2025年3月舉辦

**評分標準**

初賽：

* 35% 應用性：應用軟體創作解決相關領域之問題。
* 35% 原創性：作品的獨特性應用之創意。
* 20% 專業性：程式寫作過程是否結合運算思維和軟體開發方法與工具。
* 10% 完成性：設計架構之完整程度以及作品完成進度 。

複賽：

* 40% 設計概念：包含原創性及應用性。
* 25% 完成性：整體實作之作品是否架構完整。
* 20% 報告與展示：包含決賽文件與影片內容、現場簡報與問答。
* 15% 專業性：程式寫作過程是否結合運算思維和軟體開發方法與工具。
* 5% AI應用：(加分項目)從需求、設計、實作到測試，如何妥善應用AI協助。

**第一步：上傳圖片並辨識社區目標位置**

1. **核心功能描述**：
   * 使用者上傳一張圖片，系統分析圖片內容，辨識該圖片中的地點或社區位置。
2. **技術需求**：
   * **圖片處理與分析**：
     + 使用深度學習技術進行目標辨識（例如：YOLO、Faster R-CNN）。
     + 使用現有的 API 或模型（如 Google Vision API、OpenCV + 自訓練模型）。
   * **資料準備**：
     + 準備包含社區圖片和名稱的數據集，進行模型訓練或微調。
   * **前端界面**：
     + 設計簡易的上傳介面，接收圖片並顯示處理結果。
3. **工具與框架建議**：
   * **框架**：TensorFlow、PyTorch（模型訓練與推論）。
   * **API**：Google Vision、Microsoft Azure Computer Vision。
   * **前端**：React/Vue.js + 文件上傳插件（如 Dropzone）。

**第二步：連結實價登錄資料**

1. **核心功能描述**：
   * 獲取辨識到的社區名稱，並查詢與該社區相關的實價登錄資料。
2. **技術需求**：
   * **實價登錄數據處理**：
     + 爬取或導入台灣不動產實價登錄的數據，建構一個資料庫（如 MySQL）。
   * **查詢功能**：
     + 將辨識結果作為關鍵字，查詢資料庫中匹配的房屋資訊。
   * **數據處理與顯示**：
     + 將房價資訊以清晰的方式呈現（如表格、圖表）。
3. **工具與框架建議**：
   * **後端**：使用 Node.js、Flask、Django。
   * **資料庫**：MySQL、PostgreSQL。
   * **數據視覺化**：Chart.js、D3.js。

**第三步：整合完整流程**

1. **核心功能描述**：
   * 整合圖片上傳、社區辨識、房價查詢功能，實現流暢的使用體驗。
2. **技術需求**：
   * **API 整合**：
     + 將圖片處理、辨識 API 和房價查詢服務串聯。
   * **前後端整合**：
     + 設計完整的前端介面，提供圖片上傳、辨識結果與房價顯示。
     + 後端負責圖片處理、數據查詢和結果返回。
3. **優化功能**：
   * 加入地圖顯示功能，展示社區位置（使用 Google Maps API）。
   * 支援多圖片批量上傳與處理。

**工作流程**

1. **確定社區辨識方法**：
   * 決定是否使用現成 API 或訓練自己的模型。
2. **收集數據**：
   * 準備充足的圖片和實價登錄數據。
3. **開發分階段測試**：
   * **第一階段**：完成圖片上傳與社區辨識。
   * **第二階段**：加入實價登錄查詢功能。
   * **第三階段**：整合流程並測試。
4. **優化用戶體驗**：
   * 確保界面友好，結果顯示清晰準確。
5. **準備競賽展示**：
   * 做好簡報和實際演示，展示功能的應用價值和技術難點。

**流程分解：**

1. **圖片上傳（輸入）**
   * 使用者上傳圖片作為輸入，圖片中包含目標社區。
2. **圖片辨識（處理）**
   * 系統將圖片送入 AI 模型或 API，辨識圖片中社區的名稱或地點。
3. **資料查詢（處理）**
   * 系統使用社區名稱作為關鍵字，查詢實價登錄資料庫，篩選出相關房價資訊。
4. **結果顯示（輸出）**
   * 顯示該社區位置與相關房價數據，例如房屋均價、最高價、最低價等。

以下是針對您的需求設計的圖片辨識與房價查詢功能的流程圖，涵蓋資料的輸入與輸出，以及模組架構：

**流程分解：**

1. **圖片上傳（輸入）**
   * 使用者上傳圖片作為輸入，圖片中包含目標社區。
2. **圖片辨識（處理）**
   * 系統將圖片送入 AI 模型或 API，辨識圖片中社區的名稱或地點。
3. **資料查詢（處理）**
   * 系統使用社區名稱作為關鍵字，查詢實價登錄資料庫，篩選出相關房價資訊。
4. **結果顯示（輸出）**
   * 顯示該社區位置與相關房價數據，例如房屋均價、最高價、最低價等。

**資料需求**

1. **輸入的資料**：
   * 使用者上傳的圖片（包含社區或建築物）。
2. **圖片辨識模組所需資料**：
   * **社區圖片數據**：
     + 包含各地社區的照片、名稱，用於模型訓練或 API 的參考資料。
     + 資料來源：
       - 自行拍攝社區圖片（與模型匹配）。
       - 開放數據集（若適用）。
   * **辨識模型**：
     + 預訓練模型（如 YOLO、ResNet）。
     + Google Vision API 或其他商業解決方案。
3. **房價查詢模組所需資料**：
   * **不動產實價登錄資料**：
     + 台灣政府開放資料集（實價登錄平台）。
     + 數據清理後存入資料庫（MySQL、PostgreSQL）。
4. **輔助數據（選擇性）**：
   * 地理資訊（如 Google Maps API 提供的地圖座標）。

**輸出內容**

1. **圖片辨識的結果**：
   * 社區名稱。
   * 地點座標（選擇性）。
2. **房價數據**：
   * 平均房價。
   * 價格區間（最高價和最低價）。
   * 銷售日期與交易量。
3. **完整顯示**：
   * 社區名稱 + 房價資訊 + 地圖顯示位置。

**擴展功能**

* **推薦系統**：根據房價範圍推薦其他類似社區。
* **即時地圖展示**：將房價結果標記在地圖上。
* **歷史房價走勢**：加入圖表顯示特定社區的歷史房價變化。

一張含有 文字, 螢幕擷取畫面, 數字, 圖表 的圖片

自動產生的描述