

# 網際網路程式設計

專題名稱:新聞網頁

小組編號：Group 12

組員姓名與學號：

陳信楷 81405134

黃俊豪 81405139

鄭博文 81405142

賴國祥 81405147

陳世和 81405158

GitHub link: <https://github.com/81405139-dev/django12-.git>

Course Name / 課程名稱：網際網路程式設計

Instructor / 授課老師：王元吉

Submission Date / 繳交日期：1/4

## Abstract / 摘要

說明功能與用途：

本專題開發一個功能完整的「新聞入口網站」，旨在透過實作理解 Django 的 **MVT (Model-View-Template)** 架構。系統提供新聞瀏覽、分類篩選、全文檢索及後台管理功能。前端介面採用現代化響應式設計，並整合了側邊欄導覽與初學者開發筆記。

## 一、Introduction / 專題介紹

本專題旨在建立一個輕量化、易於導覽的新聞入口平台。透過 Django 的後端架構，實現內容分類與使用者介面的整合，讓初學者能理解網頁框架如何處理資料顯示與前端版面配置。

主要功能四大亮點：

- 動態分類導覽：** 透過頂端與側邊導航欄，使用者可快速篩選感興趣的新聞領域。
- 全文檢索系統：** 整合 Django 的 Q 物件，支援標題與內文的模糊關鍵字搜尋。
- 權限管理機制：** 只有登入後的使用者可撰寫新聞，且系統會自動關聯當前帳號為作者。
- 響應式 UI 設計：** 採用現代化簡潔版面，並設有初學者筆記與引導功能。

## 二、Model Design / 資料表設計

### Article (文章模型)

title: 新聞標題 (CharField)。

body: 新聞內文 (TextField)。

`date`: 發布時間，系統自動生成 (`DateTimeField`)。

`category`: 關聯至分類表 (`ForeignKey`)。

`author`: 關聯至使用者，支援連動刪除 (`ForeignKey, CASCADE`)。

### **Category (分類模型)**

`name`: 儲存如「校園」、「科技」等分類標籤，供前端篩選使用。

為什麼這樣設計：

**實現「一對多」的靈活管理 \* 分類與文章：** 將「分類」獨立成表，可以避免在每篇文章中重複輸入文字，且未來要新增分類（如：從「科技」增加到「AI」）時，只需在後台增加一筆分類資料，不需改動程式碼。

**作者與文章：** 連結 `User` 模型，能確保每篇文章都有明確的來源歸屬，並支援「連動刪除 (CASCADE)」，維持資料庫的整潔。

**提升開發效率與準確性 \* 自動化欄位：** 使用 `auto_now_add` 讓系統自動記錄發布時間，避免人為輸入錯誤，確保新聞的時效性真實可靠。

**ORM 技術：** 透過 Django 模型設計，開發者不需寫複雜的 SQL 指令，就能快速進行關鍵字搜尋與分類篩選。

**系統安全性與權限控管 \***  這樣的設計讓我們能輕鬆實作「只有這篇文章的作者才能修改內容」的邏輯，將使用者身分與內容資料緊密結合，防止資料被隨意竊改。

**未來的擴充空間 \***  獨立的資料表結構讓系統具備極佳的擴充性。例如：未來想加入「文章封面圖」或「閱讀次數統計」，只需在現有的 `Article` 模型中增加一個欄位，即可在不破壞現有功能的情況下完成升級。

## **三、CRUD Functions / CRUD 功能說明**

**Create (新增)：** 使用者透過前台表單發布新聞，系統自動補抓作者身分。

**Read (讀取)：** 提供列表頁面與詳細內容頁面，並具備分頁處理。

**Update (修改)：** 作者可針對已發布內容進行即時修正。

**Delete (刪除)：** 提供刪除功能，並設定

`on_delete=models.CASCADE` 以維護資料庫連動一致性。

## 四、System Pages / 系統頁面說明

### 1. 首頁與文章列表頁 (**Home & Article List Page**)

- **用途：** 這是網站的門面，展示所有最新的新聞列表，讓讀者能快速瀏覽今日焦點。
- **核心功能：**
  - **動態列表：** 系統從資料庫抓取 Article 資料，並依照發布時間倒序排列（最新文章在前）。
  - **分類篩選：** 透過頂部導航欄，使用者可點擊「校園」、「科技」等標籤，網址會帶入 `?category=ID` 參數，實現精準的文章過濾。
  - **全文搜尋：** 提供搜尋框，支援關鍵字檢索，讓使用者能從大量新聞中找到特定主題。

### 2. 文章詳細內容頁 (**Article Detail Page**)

- **用途：** 當使用者點擊列表中的標題時，會導向此頁面閱讀完整的新聞內容。
- **核心功能：**
  - **完整呈現：** 顯示文章標題、完整內文、發布日期以及作

者資訊。

- 路徑導航：頁面頂端提供「返回列表」連結，提升使用者瀏覽體驗。
- 資料抓取：透過 URL 中的 pk (Primary Key) 唯一識別碼，精準從資料庫讀取該篇新聞。

### 3. 撰寫新聞頁 (Create Article Page)

- 用途：提供給登入後的「特約記者」或管理員發布新資訊的介面。
- 核心功能：
  - 安全攔截：整合 LoginRequiredMixin，若未登入的使用者嘗試存取此網址，系統會自動導向登入頁面。
  - 自動關聯：使用者僅需填寫標題、分類與內容，系統後端會自動將「當前登入帳號」存為該文章的作者，無需手動選擇。

### 4. 編輯與修正頁 (Edit Article Page)

- 用途：允許作者對已發布的新聞進行勘誤或內容補充。
- 核心功能：
  - 預載資料：進入頁面時，系統會自動將資料庫中的原有內容填入表單。

- 成功跳轉：修正完成後，系統會自動導向回該文章的「詳細內容頁」，讓作者即時確認修改結果。

## 5. 刪除確認頁 (Delete Confirmation Page)

- 用途：執行刪除動作前的最後防線，防止誤刪重要新聞。
- 核心功能：
  - 二次確認：顯示「您確定要刪除這篇文章嗎？」的提示。
  - 連動處理：確認刪除後，資料庫會永久移除該筆記錄。  
由於模型設定了 `on_delete=models.CASCADE`，系統能確保資料關聯的完整性。

## 五、Screenshots / 系統畫面截圖



## 六、Team Contribution / 小組分工

後端開發 Django 模型架構設計與資料庫管理與網站路徑導向測試：  
陳信楷、黃俊豪、鄭博文、陳世和

文件企劃報告撰寫、簡報製作與系統文件整理：賴國祥、陳世和

## 七、Conclusion / 結論

### 1. 專題成果總結

本專題成功建構了一個基於 Django 框架的「新聞入口網站系統」。

從初期的資料庫模型設計（Model）、中間的業務邏輯處理（View），

到最終的前端頁面呈現（Template），完整實踐了 MVT 開發模式。

系統不僅達成了新聞分類、全文檢索等核心讀取功能，更進一步整

合了安全權限機制，實現了具備身分驗證的文章 CRUD（增刪改

查）功能。

### 2. 技術成長與收穫

透過本專案的開發，小組成員在以下技術領域獲得了實質的成長：

- **後端架構：** 掌握了 Django 通用類別視圖 (CBV) 的應用，理解了如何透過繼承 Mixin 快速建立安全的管理介面。
- **資料查詢：** 學習使用 Q 物件進行複雜的資料庫篩選，解決了多欄位模糊搜尋的技術挑戰。

- **前端整合：**練習將靜態 HTML/CSS 與 Django 模板語言 (DTL) 結合，實現了動態資料渲染。

### 3. 開發挑戰與對策

在開發過程中，我們遇到了「搜尋功能參數傳遞」與「模板路徑配置」等邏輯錯誤。透過查閱官方文件、解析 Django 報錯訊息 (Debug Page) 以及團隊討論，我們學會了如何系統性地進行排錯 (Troubleshooting)，這對於未來開發更大型的網際網路程式具有極大的幫助。

### 4. 未來展望 (Future Work)

雖然目前系統已具備核心功能，但仍有許多優化空間，未來我們預計朝以下方向進階：

- **多媒體支援：**加入 ImageField 支援新聞首圖上傳，提升視覺吸引力。
- **互動系統：**建立留言板功能，增加讀者與作者之間的互動。
- **性能優化：**引入快取機制 (Caching) 與非同步任務處理，以應對大量存取時的伺服器壓力。

## 八、References / 參考資料

王元吉老師，網際網路程式設計課程講義。

用途：參考 Django 環境建置、MVT 架構實作及 CRUD 邏輯開發流程。

王元吉老師，課堂示範專案範例程式碼。

用途：學習 Article 模型的建立方式與導引實作。

