

中國科技大學補助職員參加校外研習心得報告

這次參加「全面掌握 ASP.NET Core Razor Page 網站開發技巧」課程，對我而言是一段充實又有啟發性的學習經驗。公司目前的主要開發架構仍以傳統 ASP.NET WebForm 為主，使用 C# 搭配 SQL Server 資料庫進行 Web 應用系統開發。雖然這個架構穩定且維護多年，但隨著技術環境與開發需求的改變，越來越多功能需要快速調整與模組化設計，而傳統架構在這方面的彈性相對不足。因此，這次能有機會學習 ASP.NET Core 與 Razor Pages 對我來說是一個很好的契機，讓我了解新世代 Web 架構的運作模式與開發思維。

課程一開始，講師帶領我們回顧 ASP.NET Core 的整體架構，並說明它與傳統 WebForm 或 MVC 的差異。Razor Pages 採用「頁面導向」(Page-Based) 設計，每個頁面對應一個 PageModel，讓程式邏輯與介面設計分工清晰卻又緊密結合。這種結構的最大優點在於開發人員能專注於單一頁面的資料流程，而不必像 MVC 那樣頻繁切換 Controller 與 View，開發效率顯著提升。在實作階段，講師示範了如何建立資料輸入頁面與表單綁定。以往在 WebForm 裡，我們常用控制項（如 TextBox、DropDownList）加上事件處理器（例如 Button_Click），再透過後端撰寫大量的資料驗證與 SQL 操作程式碼。相對之下，Razor Pages 的 Model Binding 機制可讓表單資料自動對應至後端物件屬性，大幅簡化程式邏輯。這種設計讓我感受到現代化開發在「可讀性」與「維護性」上的進步。

課程也涵蓋了 Entity Framework Core (EF Core) 的資料存取方式。透過實作 CRUD (新增、修改、刪除、查詢) 範例，我體驗到使用 EF Core 進行資料存取不僅能減少手動撰寫 SQL 的負擔，還能透過物件導向的方式更直觀地處理資料，減少資料庫操作的錯誤機率。這對我們公司目前的 ADO.NET 寫法是很大的對比。此外，課程中也介紹了 身份驗證與授權 (Authentication & Authorization) 的設計。ASP.NET Core Identity 框架可快速實現登入、登出、權限控管等功能，且支援角色分級與資料表整合。相較於我們目前以自訂帳號驗證邏輯為主的方式，這套機制更標準化、安全性也更高。

課堂上老師也示範如何透過開發工具 visual studio2022 版本可以結合目前時下最流行的 AI 工具 (GitHub copilot) 進行程式碼開發。這部分對我影響很大，因為在現有專案中，某些功能程式碼複雜或需要排除錯誤時，Copilot 可以提供程式碼建議或自動生成模板，協助快速完成較繁瑣的程式段落，也能有效幫助檢查程式碼邏輯與正確性。這個工具的學習讓我意識到，即使短時間無法全面導入新技術，也能透過 AI 與版本控制工具輔助現有開發流程，提高效率並減少開發困難。

整體而言，這四天的學習讓我不只是了解一個新框架，更重要的是體會到 ASP.NET Core 在架構設計、效能優化、安全機制與維護便利性上的全面進化。這些知識讓我重新思考未來公司系統的發展方向，也為個人開發技能帶來了很大的提升。

備註：

- 一、研習心得報告請用電腦繕打。
- 二、研習結案報告請先上傳（校園入口網→其它類 E 化系統→研討會心得上傳），連同補助職員參加校外研習申請表、研習相關資料影本(4頁以上)及研習心得報告，並經主管簽章後，送人事室核銷。

報告人簽章	單位主管簽章	人事室主任簽章
年 月 日	年 月 日	年 月 日

