113年智慧行動商務系統開發班第1期

伺服端程式設計與開發學習評量

|  |  |  |  |
| --- | --- | --- | --- |
| **評量編號** | 2-5-1 | | |
| **評量主題** | 伺服端網站應用程式綜合實作 | **實施時間** | 2025/1/6 |
| **評量形式** | 作業紀錄 | **評量日期** | 2025/1/20 |
| **評量目的** | 評定學員是否具備伺服端程式設計能力，包含資料庫的設計與建置、使用ASP.net Core MVC進行程式功能的開發，並對資料庫進行CRUD之操作，且可設計適當的使用者介面及流程，提供適切的操作方式。 | | |
| **評量說明** | **任務一**：請依題目要求方式建立符合需求的模型類別、DbContext及資料庫。  **任務二**：請依題目要求設計前台的各項功能。  **任務三**：必須合於一般使用經驗及貼近實務的，包括資料存儲內容、規則及使用流程。 | | |
| **提交要求** | 繳交時請以FTP上傳**相關程式檔案**與**作業文件檔**各一份，**相關程式檔案**請另放置在「完成結果檔」資料夾內，最外層資料夾名稱請用**兩碼學號與姓名**，例如1號王小明，資料夾請命名為「01王小明」。 | | |

座號：15 姓名：陳延銓

**請建立「使用者端留言討論區」功能。**

**任務一：請依據題題上的需求，以Code-First方式建立符合下列需求的模型類別、DbContext及資料庫，必須包含：**

1. 主文資料表(編號(P.K)、主題、發表內容、照片、照片類型、發表人、張貼時間)
2. 回覆內容資料表(編號(P.K)、回覆內容、回覆人、回覆時間)
3. 主文資料表與回覆內容資料表之間具關聯，一則主文可以有很多則回覆內容。
4. 專案名稱、資料庫及資料表名稱可自訂。
5. 各類別中須設計相對應屬性資料型態(Data Type)及驗證規則。
6. 須製作初始化時之種子資料(Seed Data)若干筆。

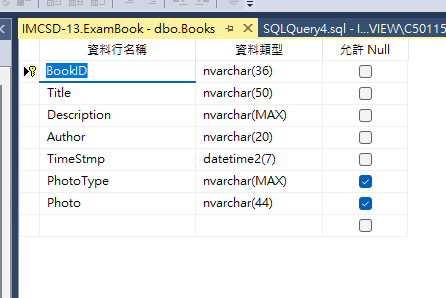
**任務二：請設計前台(使用者端)功能，必須包含：**

1. 使用者可發表新文章，資料儲存於主文資料表
2. 使用者發表文章時須填寫主題、發表內容、發表人欄位，而照片可有可無。
3. 主畫面呈現各討論文章主題，並依發表時間的新到舊排列。
4. 點選主題後可進入的該文章的討論畫面，可看到主文內容、照片及回覆內容。
5. 使用者可在文章討論畫面回覆該文章。
6. 回覆文章呈現時須依發表時間的新到舊排列。

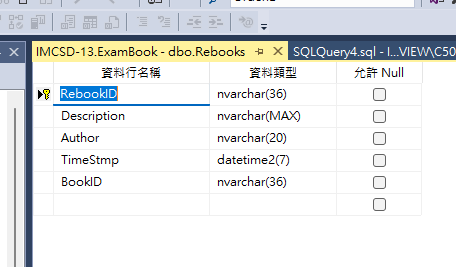
**任務三：必須合於一般使用經驗及貼近實務的，包括資料存儲內容、規則及使用流程。**

1. 資料庫(SQL Server)部份：
2. 所有資料表的設計畫面(截圖)

主文資料表設計

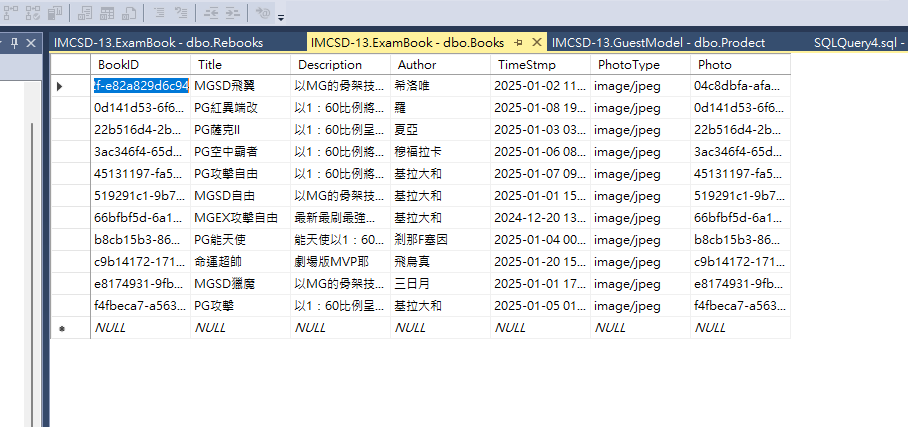
****

**回文資料表設計**

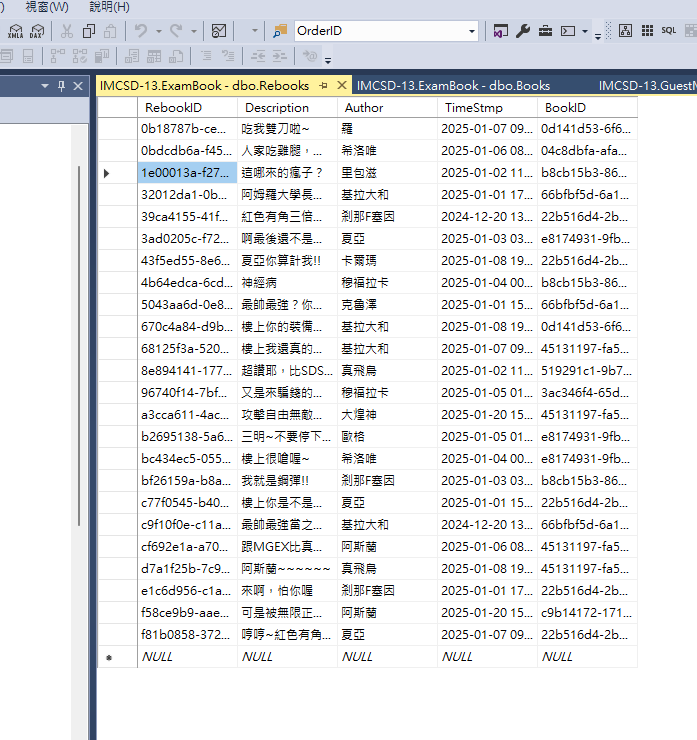
****

1. 所有資料表的資料檢視(截圖)

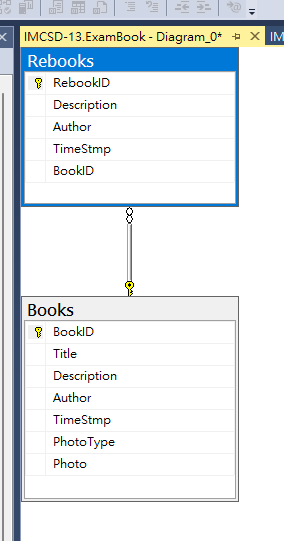
主文資料表：



回文資料表：



資料關聯圖(截圖)



1. 程式碼部份：
2. 所有的資料模型類別：

主文資料模型類別：

using System.ComponentModel.DataAnnotations;

using Microsoft.AspNetCore.Mvc;

namespace CodeFirstModelExam.Models

{

public class Book

{

[Key]

[StringLength(36, MinimumLength = 36)]

public string BookID { get; set; } = null!;

[Display(Name = "主題")]

[StringLength(50, MinimumLength = 1, ErrorMessage ="主題限制字數為1~50個字")]

[Required(ErrorMessage = "必填")]

public string Title { get; set; } = null!;

[Required(ErrorMessage = "必填")]

[Display(Name = "內容")]

[DataType(DataType.MultilineText)]

public string Description { get; set; } = null!;

[Display(Name = "作者")]

[StringLength(20, ErrorMessage = "作者最多20字")]

[Required(ErrorMessage = "必填")]

public string Author { get; set; } = null!;

[Display(Name = "發佈時間")]

[DataType(DataType.DateTime)]

[DisplayFormat(DataFormatString = "{0:yyyy/MM/dd hh:mm:ss}")]

[HiddenInput]

public DateTime TimeStmp { get; set; }

[HiddenInput]

public string? PhotoType { get; set; }

[Display(Name = "照片")]

[StringLength(44)]

public string? Photo { get; set; }

}

}

回文資料類別

using System.ComponentModel.DataAnnotations;

using Microsoft.AspNetCore.Mvc;

namespace CodeFirstModelExam.Models

{

public class Rebook

{

[Key]

[StringLength(36, MinimumLength = 36)]

public string RebookID { get; set; } = null!;

[Required(ErrorMessage = "回覆內容必填")]

[Display(Name = "回覆內容")]

[DataType(DataType.MultilineText)]

public string Description { get; set; } = null!;

[Display(Name = "作者")]

[StringLength(20, ErrorMessage = "作者最多20字")]

[Required(ErrorMessage = "必填")]

public string Author { get; set; } = null!;

[Display(Name = "回文時間")]

[DataType(DataType.DateTime)]

[DisplayFormat(DataFormatString ="{0:yyyy/MM/dd hh:mm:ss.fff}")]

[HiddenInput]

public DateTime TimeStmp { get; set; }

public string BookID { get; set; } = null!;

public virtual Book? Books { get; set; }

}

}

1. DbContext類別：

using Microsoft.EntityFrameworkCore;

namespace CodeFirstModelExam.Models

{

public class ExamBookContext : DbContext

{

public ExamBookContext(DbContextOptions<ExamBookContext> options) : base(options) { }

public virtual DbSet<Book> Books { get; set; }

public virtual DbSet<Rebook> Rebooks { get; set; }

}

}

1. Program.cs：

using CodeFirstModelExam.Models;

using Microsoft.EntityFrameworkCore;

var builder = WebApplication.CreateBuilder(args);

// Add services to the container.

builder.Services.AddControllersWithViews();

builder.Services.AddDbContext<ExamBookContext>(options =>

options.UseSqlServer(builder.Configuration.GetConnectionString("dbExamBookConnection")));

var app = builder.Build();

using (var scope = app.Services.CreateScope())

{

var service = scope.ServiceProvider;

SeedData.Initialize(service);

}

// Configure the HTTP request pipeline.

if (!app.Environment.IsDevelopment())

{

app.UseExceptionHandler("/Home/Error");

}

app.UseStaticFiles();

app.UseRouting();

app.UseAuthorization();

app.MapControllerRoute(

name: "default",

pattern: "{controller=Home}/{action=Index}/{id?}");

app.Run();

1. appsettings.json：

{

"Logging": {

"LogLevel": {

"Default": "Information",

"Microsoft.AspNetCore": "Warning"

}

},

"AllowedHosts": "\*",

"ConnectionStrings": {

"dbExamBookConnection": "Data Source=IMCSD-13; Database=ExamBook;TrustServerCertificate=True;User ID = karl; Password=pid/900af"

}

}

1. 所有Controller(請標示檔名)

BooksController.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using Microsoft.AspNetCore.Mvc;

using Microsoft.AspNetCore.Mvc.Rendering;

using Microsoft.EntityFrameworkCore;

using CodeFirstModelExam.Models;

namespace CodeFirstModelExam.Controllers

{

public class BooksController : Controller

{

private readonly ExamBookContext \_context;

public BooksController(ExamBookContext context)

{

\_context = context;

}

// GET: Books

public async Task<IActionResult> Index()

{

var book = await \_context.Books.OrderByDescending(t => t.TimeStmp).ToArrayAsync();

return View(book);

}

// GET: Books/Details/5

public async Task<IActionResult> Content(string id)

{

if (id == null)

{

return NotFound();

}

var book = await \_context.Books

.FirstOrDefaultAsync(m => m.BookID == id);

if (book == null)

{

return NotFound();

}

return View(book);

}

// GET: Books/Create

public IActionResult Create()

{

return View();

}

// POST: Books/Create

// To protect from overposting attacks, enable the specific properties you want to bind to.

// For more details, see http://go.microsoft.com/fwlink/?LinkId=317598.

[HttpPost]

[ValidateAntiForgeryToken]

public async Task<IActionResult> Create([Bind("BookID,Title,Description,Author,TimeStmp,PhotoType,Photo")] Book book, IFormFile? newPhoto)

{

book.TimeStmp = DateTime.Now;

string descript = book.Description;

book.Description = descript.Replace("\r\n", "<br>");

if (newPhoto != null && newPhoto.Length > 0)

{

if (newPhoto.ContentType != "image/jpeg" && newPhoto.ContentType != "image/png")

{

ViewData["ErrMsg"] = "請上傳正確JPG或PNG格式";

return View(book);

}

string subFileName = newPhoto.ContentType.Substring(6, newPhoto.ContentType.Length - 6);

string fileName = book.BookID + "." + subFileName;

//string fileName = book.BookID + ".jpg";

string BookPhotoPath = Path.Combine(Directory.GetCurrentDirectory(), "wwwroot", "BookPhotos", fileName);

using (FileStream stream = new FileStream(BookPhotoPath, FileMode.Create))

{

newPhoto.CopyTo(stream);

}

book.Photo = fileName;

book.PhotoType = newPhoto.ContentType;

}

if (ModelState.IsValid)

{

\_context.Add(book);

await \_context.SaveChangesAsync();

return RedirectToAction(nameof(Index));

}

return View(book);

}

private bool BookExists(string id)

{

return \_context.Books.Any(e => e.BookID == id);

}

}

}

RebooksController.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using Microsoft.AspNetCore.Mvc;

using Microsoft.AspNetCore.Mvc.Rendering;

using Microsoft.EntityFrameworkCore;

using CodeFirstModelExam.Models;

namespace CodeFirstModelExam.Controllers

{

public class RebooksController : Controller

{

private readonly ExamBookContext \_context;

public RebooksController(ExamBookContext context)

{

\_context = context;

}

// GET: Rebooks/Create

public IActionResult Create(string BookID)

{

ViewData["BookID"] = BookID;

return View();

}

// POST: Rebooks/Create

// To protect from overposting attacks, enable the specific properties you want to bind to.

// For more details, see http://go.microsoft.com/fwlink/?LinkId=317598.

[HttpPost]

[ValidateAntiForgeryToken]

public async Task<IActionResult> Create([Bind("RebookID,Description,Author,TimeStmp,BookID")] Rebook rebook)

{

rebook.TimeStmp = DateTime.Now;

if (ModelState.IsValid)

{

\_context.Add(rebook);

await \_context.SaveChangesAsync();

return Json(rebook);

}

ViewData["BookID"] = rebook.BookID;

return Json(rebook);

}

// GET: Rebooks/Edit/5

public async Task<IActionResult> Edit(string id)

{

if (id == null)

{

return NotFound();

}

var rebook = await \_context.Rebooks.FindAsync(id);

if (rebook == null)

{

return NotFound();

}

ViewData["BookID"] = new SelectList(\_context.Books, "BookID", "BookID", rebook.BookID);

return View(rebook);

}

public IActionResult GetRebookByViewComponent(string id)

{

return ViewComponent("VCReBooks", new { bookID = id });

}

}

}

VCRebooks.cs

using CodeFirstModelExam.Models;

using Microsoft.AspNetCore.Mvc;

using Microsoft.EntityFrameworkCore;

namespace CodeFirstModelExam.ViewComponents

{

public class VCReBooks:ViewComponent

{

private readonly ExamBookContext \_context;

public VCReBooks(ExamBookContext context) { \_context = context; }

public async Task<IViewComponentResult> InvokeAsync(string bookID)

{

var db = await \_context.Rebooks.Where(t => t.BookID == bookID).OrderByDescending(t => t.TimeStmp).ToArrayAsync();

return View(db);

}

}

}

1. 執行結果部份：

請分別截取各種功能的展現畫面至少5張圖：

