114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** | 2025/7/14 |
| **評量形式** | 作業紀錄 | **評量日期** | 2025/8/15 |
| **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： 05 姓名： 鄭冠怡

**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：**

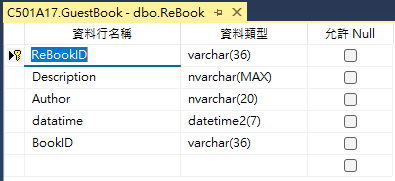
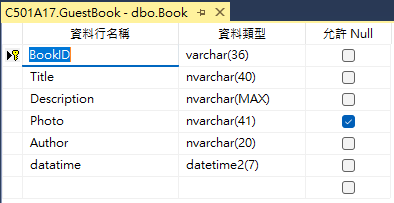
1. 主文資料表(編號(P.K)、主題、發表內容、照片、照片類型、發表人、張貼時間)
2. 回覆內容資料表(編號(P.K)、回覆內容、回覆人、回覆時間)
3. 主文資料表與回覆內容資料表之間具關聯，一則主文可以有很多則回覆內容。
4. 專案名稱、資料庫及資料表名稱可自訂。
5. 各類別中須設計相對應屬性資料型態(Data Type)及驗證規則。
6. 須製作初始化時之種子資料(Seed Data)若干筆。

**任務二：請設計前台(使用者端)功能，必須包含：**

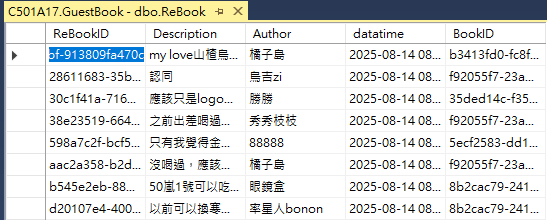
1. 使用者可發表新文章，資料儲存於主文資料表
2. 使用者發表文章時須填寫主題、發表內容、發表人欄位，而照片可有可無。
3. 主畫面呈現各討論文章主題，並依發表時間的新到舊排列。
4. 點選主題後可進入的該文章的討論畫面，可看到主文內容、照片及回覆內容。
5. 使用者可在文章討論畫面回覆該文章。
6. 回覆文章呈現時須依發表時間的新到舊排列。

**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。**

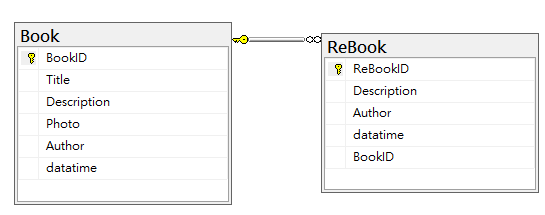
1. 資料庫(SQL Server)部份：
2. 所有資料表的設計畫面(截圖)



1. 所有資料表的資料檢視(截圖)

資料關聯圖(截圖)



1. 程式碼部份：
2. 所有的資料模型類別：

--Book.cs

using Microsoft.AspNetCore.Mvc;

using System.ComponentModel.DataAnnotations;

namespace GuestBooks.Models

{

public partial class Book

{

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

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

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

public string? Photo { get; set; }

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

public DateTime CreateDate { get; set; } = DateTime.Now;

public virtual List<ReBook>? ReBooks { get; set; }

}

}

-- ReBook.cs

using Microsoft.AspNetCore.Mvc;

using System.ComponentModel.DataAnnotations;

using System.ComponentModel.DataAnnotations.Schema;

namespace GuestBooks.Models

{

public partial class ReBook

{

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

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

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

public DateTime CreateDate { get; set; } = DateTime.Now;

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

public Book? Book { get; set; }

--MetaData.cs

using Microsoft.AspNetCore.Mvc;

using System.ComponentModel.DataAnnotations;

using System.ComponentModel.DataAnnotations.Schema;

namespace GuestBooks.Models

{

public class BookData

{

[Key]

[Display(Name = "留言編號")]

[StringLength(36)]

[HiddenInput]

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

[Display(Name = "發表主題")]

[StringLength(40, MinimumLength = 3, ErrorMessage = "主題長度需介於3~40個字")]

[Required(ErrorMessage = "發表主題必填")]

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

[Display(Name = "發表內容")]

[DataType(DataType.MultilineText)]

[Required(ErrorMessage = "發表內容必填")]

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

[Display(Name = "照片")]

[StringLength(41)] //36碼GUID+.JPG/.PNG/.JPEG(因為照片副檔名最長為5個字元)

public string? Photo { get; set; }

[Display(Name = "發表人名稱")]

[StringLength(20, ErrorMessage = "發表人名稱最多20個字")]

[Required(ErrorMessage = "發表人名稱必填")]

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

[Display(Name = "發布時間")]

[DataType(DataType.DateTime)]

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

[HiddenInput]

public DateTime CreateDate { get; set; } = DateTime.Now;

}

public class ReBookData

{

[Key]

[Display(Name = "回覆編號")]

[StringLength(36, MinimumLength = 36)]

[HiddenInput]

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!;

[Required(ErrorMessage = "回覆人名稱必填")]

[Display(Name = "回覆人名稱")]

[StringLength(20, ErrorMessage = "回覆人名稱最多20個字")]

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

public DateTime CreateDate { get; set; } = DateTime.Now;

[ForeignKey("Book")]

[HiddenInput]

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

}

[ModelMetadataType(typeof(BookData))]

public partial class Book

{

}

[ModelMetadataType(typeof(ReBookData))]

public partial class ReBook

{

}

}

--SeedData.cs

using Microsoft.EntityFrameworkCore;

namespace GuestBooks.Models

{

public class SeedData

{

public static void Initailize(IServiceProvider serviceProvider)

{

using (GuestBookContext context = new GuestBookContext(serviceProvider.GetRequiredService<DbContextOptions<GuestBookContext>>()))

{

{

if (!context.Book.Any())

{

string[] guid = { Guid.NewGuid().ToString(), Guid.NewGuid().ToString(), Guid.NewGuid().ToString(), Guid.NewGuid().ToString(), Guid.NewGuid().ToString() };

context.Book.AddRange(

new Book

{

BookID = guid[0],

Title = "挑戰手搖飲金萱茶之巔",

Description = "金茶伍的日光金萱是我喝過最好喝的純茶類",

Author = "泉hooshi",

Photo = guid[0] + ".jpg",

CreateDate = DateTime.Now

},

new Book

{

BookID = guid[1],

Title = "很像某品牌...",

Description = "所以有人知道三分春色跟麻古有關係嗎",

Author = "glassbox666",

Photo = guid[1] + ".jpg",

CreateDate = DateTime.Now

},

new Book

{

BookID = guid[2],

Title = "總之提神一下",

Description = "我有金萱癮，一天一罐很基本吧",

Author = "迪肉燦",

Photo = guid[2] + ".jpg",

CreateDate = DateTime.Now

},

new Book

{

BookID = guid[3],

Title = "咀嚼控到處喝飲料的日記",

Description = "好喝~推薦三分春色的金萱雙Q",

Author = "民大土土",

Photo = guid[3] + ".jpg",

CreateDate = DateTime.Now

},

new Book

{

BookID = guid[4],

Title = "支持山楂烏龍統治地球",

Description = "茶摩什麼時候才要全通路販售這款山楂烏龍",

Author = "DrinKing",

Photo = guid[4] + ".jpg",

CreateDate = DateTime.Now

}

);

context.ReBook.AddRange(

new ReBook

{

ReBookID = Guid.NewGuid().ToString(),

Description = "沒喝過，應該不是連鎖品牌ㄅ",

Author = "橘子島",

CreateDate = DateTime.Now,

BookID = guid[0]

},

new ReBook

{

ReBookID = Guid.NewGuid().ToString(),

Description = "之前出差喝過，有股很重的奶香味",

Author = "秀秀枝枝",

CreateDate = DateTime.Now,

BookID = guid[0]

},

new ReBook

{

ReBookID = Guid.NewGuid().ToString(),

Description = "認同",

Author = "烏吉zi",

CreateDate = DateTime.Now,

BookID = guid[0]

},

new ReBook

{

ReBookID = Guid.NewGuid().ToString(),

Description = "應該只是logo跟門店設計很像而已?",

Author = "勝勝",

CreateDate = DateTime.Now,

BookID = guid[1]

},

new ReBook

{

ReBookID = Guid.NewGuid().ToString(),

Description = "只有我覺得金萱喝起來有股怪味嗎",

Author = "88888",

CreateDate = DateTime.Now,

BookID = guid[2]

},

new ReBook

{

ReBookID = Guid.NewGuid().ToString(),

Description = "以前可以換寒天，比珍珠好吃",

Author = "率星人bonon",

CreateDate = DateTime.Now,

BookID = guid[3]

},

new ReBook

{

ReBookID = Guid.NewGuid().ToString(),

Description = "50嵐1號可以吃很飽",

Author = "眼鏡盒",

CreateDate = DateTime.Now,

BookID = guid[3]

},

new ReBook

{

ReBookID = Guid.NewGuid().ToString(),

Description = "my love山楂烏龍^3^",

Author = "橘子島",

CreateDate = DateTime.Now,

BookID = guid[4]

}

);

context.SaveChanges();

string SeedPhotoPath = Path.Combine(Directory.GetCurrentDirectory(), "SeedPhoto");

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

string[] files = Directory.GetFiles(SeedPhotoPath);

for (int i = 0; i < files.Length; i++)

{

string destFile = Path.Combine(BookPhotosPath, guid[i] + ".jpg");

File.Copy(files[i], destFile);

}

}

}

}

}

}

}

1. DbContext類別：

using Microsoft.EntityFrameworkCore;

namespace GuestBooks.Models

{

public class GuestBookContext : DbContext

{

public GuestBookContext(DbContextOptions<GuestBookContext> options)

: base(options)

{

}

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

public virtual DbSet<ReBook> ReBook { get; set; }

protected override void OnModelCreating(ModelBuilder modelBuilder)

{

modelBuilder.Entity<Book>(entity =>

{

entity.HasKey(e => e.BookID).HasName("PK\_BookID");

entity.Property(e => e.BookID).HasMaxLength(36).IsUnicode(false);

entity.Property(e => e.Title)

.HasMaxLength(40);

entity.Property(e => e.Photo)

.HasMaxLength(41);

entity.Property(e => e.Author)

.HasMaxLength(20);

entity.Property(e => e.CreateDate)

.HasColumnName("datatime");

});

modelBuilder.Entity<ReBook>(entity =>

{

entity.HasKey(e => e.ReBookID)

.HasName("PK\_ReBookID");

entity.Property(e => e.ReBookID)

.HasMaxLength(36).IsUnicode(false);

entity.Property(e => e.Author)

.HasMaxLength(20);

entity.Property(e => e.CreateDate)

.HasColumnName("datatime");

});

}

}

}

1. Program.cs：

using Microsoft.EntityFrameworkCore;

using GuestBooks.Models;

var builder = WebApplication.CreateBuilder(args);

// Add services to the container.

builder.Services.AddControllersWithViews();

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

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

//////////////////////////////////////////////////

var app = builder.Build();

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

{

var services = scope.ServiceProvider;

SeedData.Initailize(services);

}

// 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": {

"GuestBookConnection": "Data Source=C501A17;Database=GuestBook;TrustServerCertificate=True;User ID=abc456;Password=123"

}

}

1. 所有Controller(請標示檔名)

--PostBookController

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 GuestBooks.Models;

namespace GuestBooks.Controllers

{

public class PostBooksController : Controller

{

private readonly GuestBookContext \_context;

public PostBooksController(GuestBookContext context)

{

\_context = context;

}

public async Task<IActionResult> Index()

{

var result = await \_context.Book.OrderByDescending(b => b.CreateDate).ToListAsync();

return View(result);

}

public async Task<IActionResult> Details(string id)

{

if (id == null)

{

return NotFound();

}

var book = await \_context.Book

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

if (book == null)

{

return NotFound();

}

return View(book);

}

public async Task<IActionResult> Display(string id)

{

if (id == null)

{

return NotFound();

}

var book = await \_context.Book

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

if (book == null)

{

return NotFound();

}

return View(book);

}

public IActionResult Create()

{

return View();

}

[HttpPost]

[ValidateAntiForgeryToken]

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

{

book.CreateDate = DateTime.Now;

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

{

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

{

ViewData["ErrMessage"] = "照片格式錯誤，請上傳JPEG、JPG或PNG格式的圖片。";

return View();

}

string fileName = book.BookID+Path.GetExtension(newPhoto.FileName);

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

using ( FileStream fs= new FileStream(filePath, FileMode.Create))

{

newPhoto.CopyTo(fs);

}

book.Photo = fileName;

}

if (ModelState.IsValid)

{

\_context.Add(book);

await \_context.SaveChangesAsync();

return RedirectToAction(nameof(Index));

}

return View(book);

}

private bool BookExists(string id)

{

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

}

}

}

--RePostBookController

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 GuestBooks.Models;

namespace GuestBooks.Controllers

{

public class RePostBooksController : Controller

{

private readonly GuestBookContext \_context;

public RePostBooksController(GuestBookContext context)

{

\_context = context;

}

public IActionResult Create(string BookID)

{

ViewData["BookID"] = BookID;

return View();

}

[HttpPost]

[ValidateAntiForgeryToken]

public async Task<IActionResult> Create([Bind("ReBookID,Description,Author,CreateDate,BookID")] ReBook reBook)

{

if (ModelState.IsValid)

{

\_context.Add(reBook);

await \_context.SaveChangesAsync();

return Json(reBook);

}

return Json(reBook);

}

public IActionResult GetReBookByViewComponent(string BookID)

{

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

}

}

}

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：



