

國立高雄科技大學(第一校區)

資訊管理系

111 學年度第 2 學期

實務專題期末報告

基於文字生成圖像技術之繪本
創作平台-「StoryDream 故事
夢」

組員：林澤權 (C109118234)

董秉學 (C109118232)

洪偉錦 (C109118242)

林貞智 (C109118245)

指導老師：黃承龍 博士

中 華 民 國 112 年 6 月

國立高雄科技大學(第一校區)資訊管理系

大學部實務專題期末報告確認單

專題題目	基於文字生成圖像技術之繪本創作平台 - 「StoryDream 故事夢」
組員	林澤權、董秉學、洪偉錦、林貞智
指導老師	黃承龍
<p>* 專題內容如無須修正，煩請指導老師簽名，謝謝。</p> <p>指導老師簽名：黃承龍</p>	

摘要

本研究開發一個自動生成繪本的創作平台「StoryDream 故事夢」，主要客群設定為中高年級以上的學童，目的讓使用者選擇提示詞（故事情節），系統會自動產生合乎提示詞的生動圖片，並配上旁白整理成繪本。

此電子繪本創作平台，會使用到文字生成圖像技術，將其應用於創作繪本的情境，加上簡易，輔助完成一本電子繪本，不僅可讓學童學習字詞與圖像的連結性以及提供一個創作空間，也可以展示作品在平台上，吸引學童閱讀這些豐富的創作作品。

關鍵字：Generative AI、text-to-image、Diffusion model、生成圖像技術

目錄

壹、 緒論.....	5
一、研究動機與研究問題	5
貳、 文獻探討.....	6
一、文字生成圖像技術	6
二、擴散模型－Diffusion Model	6
三、文字生成圖像模型－Stable Diffusion	7
四、擴散模型框架－Diffusers	8
五、網站建構技術－Django	8
參、 研究方法及步驟.....	9
一、後端開發	9
（一）擴散模型	9
（二）資料庫	9
（三）創作繪本平台	10
二、系統架構	11
三、系統評估	11
肆、 研究結果.....	12
一、系統展示	12
（一）閱覽中心	12
（二）繪本創作	14
（三）會員機制	19
二、研究貢獻	20
三、未來展望	20
伍、 參考文獻.....	21
陸、 附件.....	22
一、本系統系統測試建置繪本	22
二、會員中心程式碼	23
三、閱覽大廳程式碼	26
四、創作中心程式碼	38
五、儲存頁面程式碼	60

壹、緒論

一、研究動機與研究問題

文字生成圖像已發展頗為成熟，是人工智慧生成內容的技術（Generative Artificial Intelligence）之一，在文字生成上有 ChatGPT，而將文字結合圖像則有文字生成圖像技術的 DALL-E 2。在文字生成圖像方面，目前市面上出現的產品除了 DALL-E 2 之外，還有 Midjourney 和 Imagen，其生成圖像的技術都來自 diffusion model，也有許多基於 diffusion model 的應用，如圖像生成圖像、圖像修復等。

Stable Diffusion 是由慕尼黑大學的 CompVis 研究團體與 Stability AI 合作開發的生成圖像模型，它以低計算量和開源碼提供全新的創作可能。使用到的 latent diffusion model 屬於 diffusion model 的變體，特色是降低訓練和生成的計算，且提供其原始碼及預訓練的權重，使得更能廣泛作為商業用途。

文字生成圖像的技術產生了各種應用情境，不過現在看到的應用大部分都以單張圖像的應用，而非多張連續且風格一致的生動圖像。於是形成了本研究的動機：若能產生風格一致的圖像，並配合旁白敘述，是不是能做成一個生動、有趣的故事？

此外，若讓學童使用或許能提高他們對於文字連結圖像的記憶性與寫作力，基於這樣的想法，延伸出想要建立一個繪本創作的平台，結合圖像生成模型的技術，不僅可以在平台上透過選擇字詞的方式組合成一本電子繪本，並在平台上提供他人閱讀，也可在平台上觀賞其他人做的繪本。

在繪本創作平台的實現上，以 Django 作為平台的網頁框架，並使用 hugging face 開發的 Diffusers 套件加入生成圖像模型—Stable Diffusion，另外需建立字詞資料庫提供使用者方便進行創作，研究問題將可分為下列三點：

1. 資料庫的建置，以儲存繪本、使用者資訊以及生圖字詞等資料。
2. 能夠透過參數的設定，產生多張具關聯性圖像的模型。
3. 將模型結合至網頁上，以實現繪本創作平台。

貳、文獻探討

一、文字生成圖像技術

文字生成圖像（text-to-image）是一種深度學習模型，能夠根據文字描述輸出與文字相匹配的圖像。這種模型的開發始於 2010 年代中期隨著深度神經網絡技術的發展而不斷進步，從 alignDRAW 模型可生成低解析度且模糊的圖像及生成對抗網路對文字生成圖像的應用，到現今廣泛的影響和應用的擴散模型可以生成較逼真且具豐富想像力的圖像。

文字生成圖像模型通常由兩個部分組成：一個語言模型，負責將輸入的文本轉換為機器可理解的表示；一個圖像生成模型，負責根據語言表示生成圖像。這兩個部分可以採用不同的架構，例如循環神經網絡、變分自編碼器、生成對抗網絡、Transformer 模型等。

另外此模型需要在大量的文字及圖像對資料集上進行訓練，通常是從網際網路上抓取來的。常用的數據集包括 COCO、Oxford-120 Flowers 和 CUB-200 Birds 等。

如今文字生成圖像模型較著名的有著 Stable Diffusion、Dall-E 2、MidJourney，可用於藝術創作、視覺設計、教育娛樂等[1]。

二、擴散模型—Diffusion Model

當前最受歡迎的文字生成圖像應用 Stable Diffusion、Dall-E 2、MidJourney 等，都是基於 Diffusion Model 模型演變。

擴散模型 Diffusion Model 的訓練過程分成兩個步驟：擴散過程及逆擴散過程。擴散過程指將原始影像資料投射到像素空間上，而逆擴散過程則從原像素空間逐步恢復至原始影像資料完成。

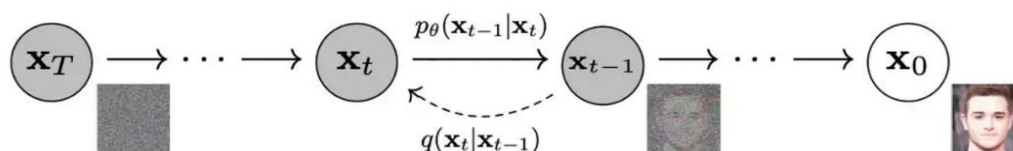


圖 1 擴散模型 - 逆擴散過程[2]

三、文字生成圖像模型—Stable Diffusion

Stable Diffusion 是一個專門用文字生成圖像的模型，基於 Diffusion Model 的變體「Latent Diffusion Model」[3]，原 diffusion model 雖然已經證明可實現生成圖像的結果。不過缺點是在訓練及推算時會消耗大量記憶體及花費許多時間，且在實際圖像空間作運算，這使得在生成高解析度影像時會變得非常昂貴[4]。

而 Latent diffusion model 是先訓練一個 AutoEncoder（包含 decoder 及 encoder）將圖像在像素空間進行壓縮，並選擇在低維度的潛在空間（latent space）上進行擴散過程降低記憶體及複雜的計算[3]。

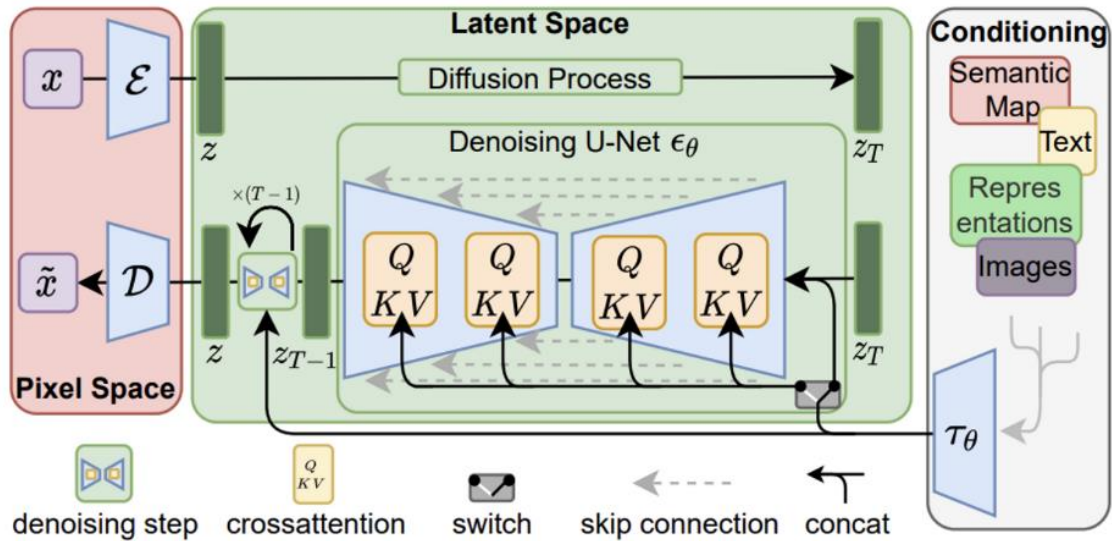


圖 2 latent diffusion 結構圖[3]

其中 Latent diffusion model 在潛在空間上的擴散過程是以 U-Net 架構實現，但重要的是引入條件機制，通過 cross-attention 方式實現多模態訓練(例如文字、圖像等)，使得除了可進行文字生成圖像任務外，還有圖像修復任務[4]，如圖 2。

Stable Diffusion 模型提供基本提示詞（prompt）輸入、圖像大小來產生圖像，也可以透過其他參數，試圖完成風格相似的圖像內容，而其參數定義如下[5]：

1. Prompt：生成圖像所提示詞，
2. Steps：圖像生成的訓練次數
3. Scale：可調整生成圖像與提示詞的相關程度
4. Seed：使用相近的種子數可獲得與之前生成的圖像相同的結果。
5. Negative Prompt：輸入反向提示詞，以避免在圖像生成中出現的特徵。

了解 Stable Diffusion 的訓練過程及參數定義後，若能善用參數的使用，對於生成連續圖像關聯性認為是有機會的，而此研究透過 Stable Diffusion 模型為基礎，認為建立電子繪本的平台是有可行性的。

四、擴散模型框架—Diffusers

Diffusers 是由 Hugging face 推出用於預訓練 Diffusion Model 的工具套件，主要滿足生成圖像的需求：選擇模型、訓練、參數設定等，而 Stable Diffusion 模型是可完全相容 Diffusers[6]。

Diffusers 具有下列幾個功能：

1. 透過較短程式碼，即可利用 Stable Diffusion 擴散模型完成生成圖像任務。
2. 可自行選擇不同的雜訊調整器（Scheduler），來平衡生成圖像模型的生成速度及品質之間的關係。
3. 可使用不同的生成模型，進行生圖任務。

五、網站建構技術—Django

本研究網站建構以 Django 作為開發環境。Django 是由 Python 程式語言所建立的網路框架，可以快速開發安全和可維護的網站，採用以類 MVC 的軟體設計模式 — MVT（模型 Model、視圖 View、樣本 Template），另外 Django 已事先提供開發網站架設的基礎功能：簡化表單建設、用戶身分驗證及後台服務應用等，另外也可以使用 python 工具套件[7]。

參、研究方法及步驟

本專案主要目標為提供一個結合文字生成圖像模型的電子繪本創作平台，此服務可使學童不受能力限制地創作故事，並加強文字及圖像的記憶力。此外可以將創作的電子繪本分享在提供的平台上。主要開發的挑戰為：降低文字生圖的條件門檻、提示詞蒐集、有效產生具關聯多張圖像、網路平台建置等等，我們開發的步驟重心分成前、後端之設計，以及前後端的整合，如圖 3：

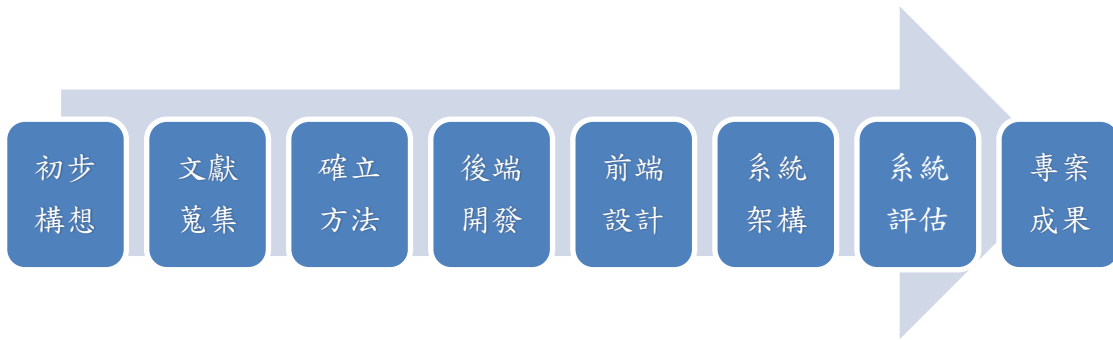


圖 3 本專案研究步驟流程圖

一、後端開發

後端開發以 Python 作為主要程式語言開發，並根據功能和實作方法不同，主要分成擴散模型導入、資料庫整理、創作繪本平台三個部分。

（一）擴散模型

選擇使用 Hugging Face 開發的 Diffuser API，以導入 Stable Diffusion 模型及提供關鍵詞輸入、種子數、圖像訓練次數以讓使用者進行繪本創作。

（二）資料庫

本平台使用 Django 框架內建的 Sqlite3 資料庫，製作下列資料庫，主要分別為提示詞資料庫、風格資料庫、會員資料庫以及繪本資料庫，UML 圖如圖 4：

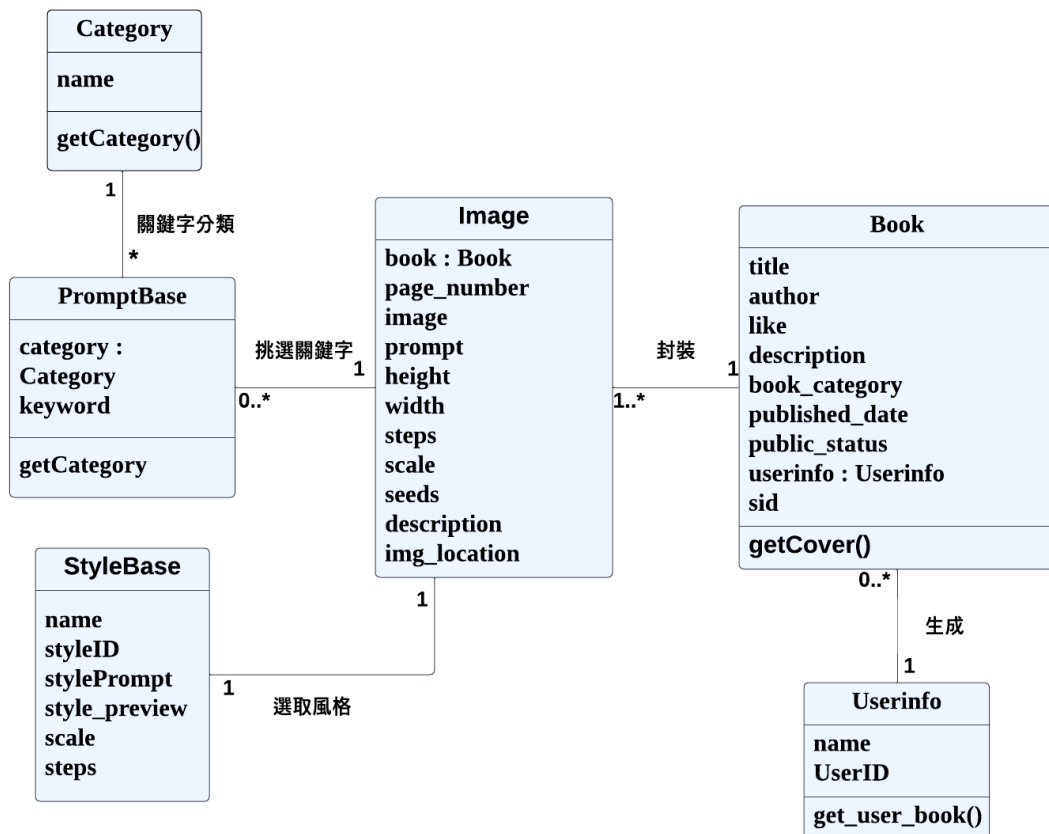


圖 4 資料庫關聯圖

1. 提示詞資料庫：將蒐集提示詞存進資料庫中，讓使用者創作繪本時，可以直接點擊提供的字詞自動輸入至提示詞中，並使用 python deep-translator 套件進行中文詞翻譯成英文，增加模型生圖時符合理想生成的機率，以利減少生圖時操作的障礙。
2. 風格資料庫：每一個風格會有穩定生成指定風格的固定提示詞，使得生圖的方向將依據該風格固有提示詞、參數設定進行生圖。
3. 繪本資料庫：提供繪本的標題、大綱、作者資訊、類型等，及連結繪本內頁的一對多圖像資料庫等屬性。
4. 使用者資料庫：蒐集使用者資訊及創作的繪本。

(三) 創作繪本平台

主要流程如圖 5，一開始將會導入風格資料庫，提供使用者選擇接下來繪本創作風格，而該風格內會固定其中的 prompt、scale、steps，

以利製作繪本的圖像將會根據風格參數設定並配合使用者輸入進行訓練，接著會進行創作，透過提示詞資料庫分類提供了使用者去選擇提示詞並搭配，降低圖像訓練的難度，最後完成繪本後儲存至系統，並讓使用者輸入繪本標題、大綱、類型及選擇公開或隱藏：



圖 5 本專案繪本系統流程圖

二、系統架構

主要以 Django 作為網站框架，將前端介面及後端資料庫及功能進行整合，以利完成專案的製作，而整個網站架構圖如圖 6。

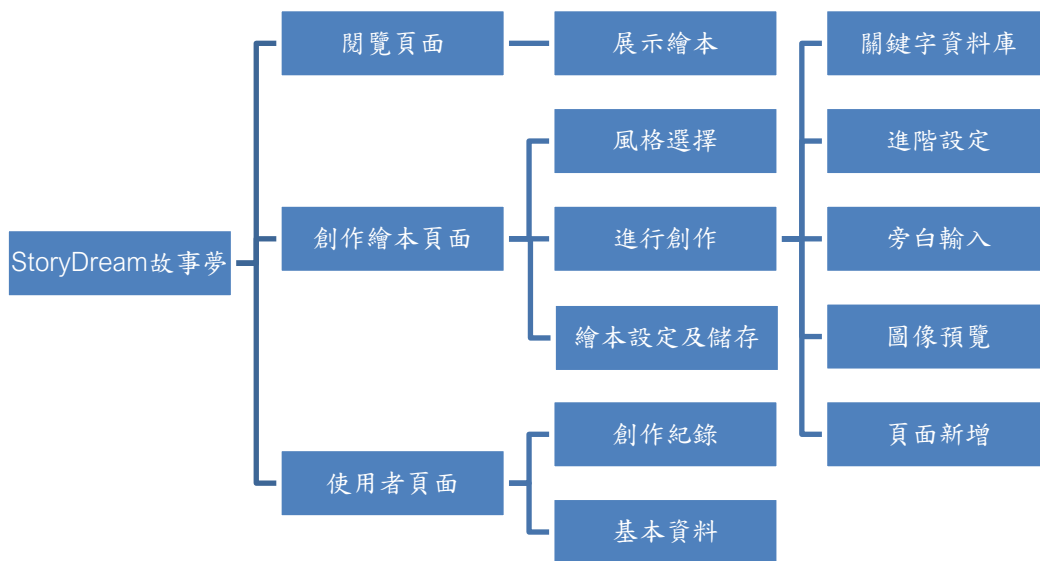


圖 6 網站架構圖

三、系統評估

為獲得更佳成果，將會找相關受測者進行測試，並填寫滿意度問卷取得回饋，以利改善網站的功能。

肆、研究結果

一、系統展示

本研究網站透過 Stable Diffusion 的文本生成圖像技術作為應用，提供中高年級學童藉由文字敘述創作出繪本，並分享出去，我們建立了關鍵字資料庫提供學童選擇，實作出多組風格相關聯性的圖像，加上文字旁白，並串聯起來，完成一本繪本。

學童們可在「StoryDream 故事夢」中，進行天馬行空的創作，完成屬於自己的繪本，而系統各頁面功能如下：

（一）閱覽中心

首頁和公開展示的作品如圖 7，手機響應式網站如圖 8：

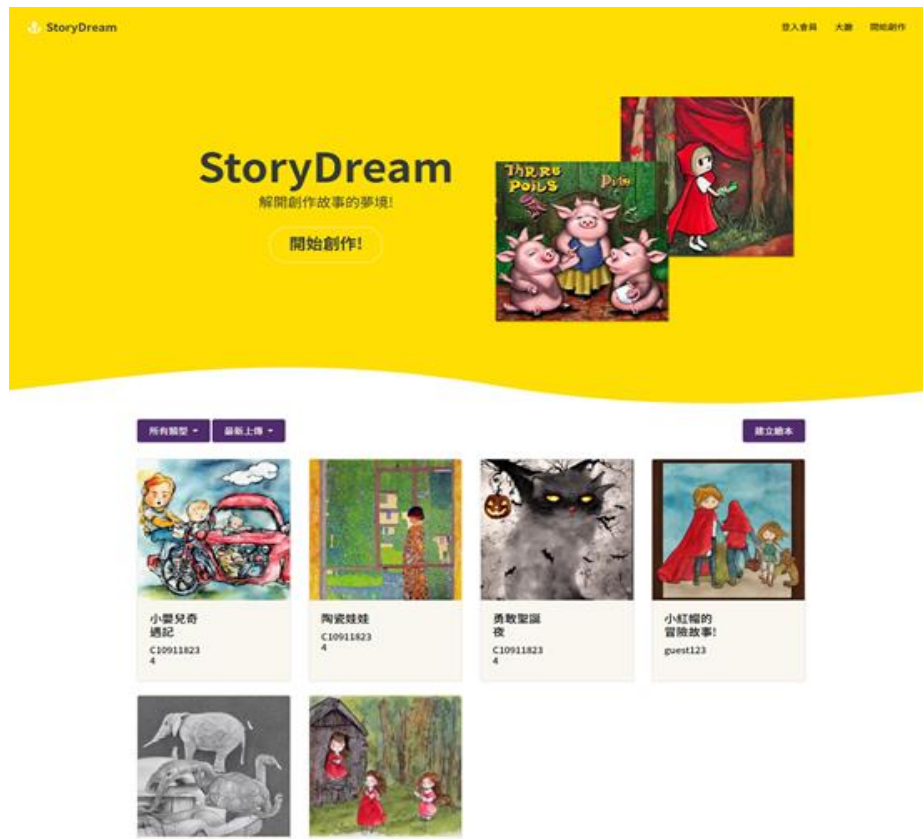


圖 7 「StoryDream 故事夢」：首頁、使用者公開展示繪本

此外，本系統也有建置響應式網頁設計，排除單一裝置操作本系統，不僅能在個人電腦上進行操作，也可透過手持式裝置（如：平板、手機等）進行閱覽及創作。



圖 8 StoryDream 網站首頁-手機畫面(本網站為響應式網站)

顯示所有創作者已公開的繪本，點擊後即可閱讀其他使用者透過本系統建置的電子繪本，供其他使用者閱讀。

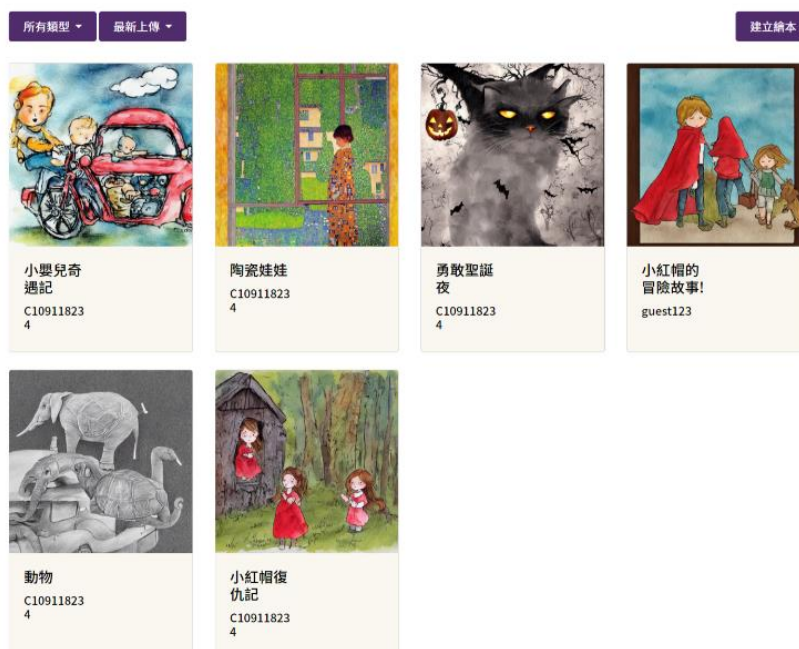


圖 9 閱覽中心公開繪本

在這些已公開繪本中，點擊進去即可觀賞選擇其他使用者透過此系統所建置的繪本作品，也可透過該網頁看到作者的資訊。

左側為作者資訊與故事大綱，右側為繪本主體，並透過下方選單切換頁數，閱讀作品其他內容，如圖 10。



圖 10 繪本閱覽頁面

(二) 繪本創作

創作過程如下，主要分成三個階段：

- 選擇繪本創作的風格主題
- 創作過程
- 繪本故事標題內容及基本設定

1. 選擇繪本創作的風格主題

使用者先透過【風格選擇頁面】選擇所要的繪本風格，接下來繪本創作的風格都會以選擇德風格作為主題。



圖 12 選擇創作風格頁面

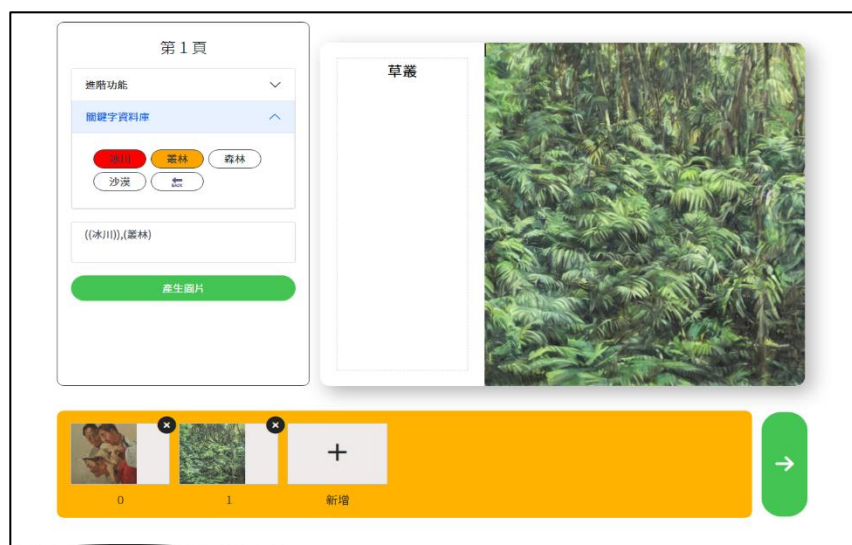


圖 11 繪本創作頁面

2. 繪本創作

選擇繪本的創作風格後，即可進行創作，創作功能如下：

1. 關鍵字資料庫

- 關鍵字預覽區：顯示生圖所運用到的提示詞，同時也可於該區進行更改，也可透過關鍵字資料庫點選。
- 關鍵字資料庫：依據分類讓使用者選擇生成圖像的提示詞(背景、時間、物件、行為等)，也可自行輸入添加至預覽區。
- 調整權重：當透過關鍵字資料庫點擊關鍵詞時，則可以依據點擊次數調整權重，或是於預覽區進行設定，每點擊一次增加一括號將對詞的關注度提高 1.1 倍，如下圖：



圖 13 關鍵字資料庫調整權重功能

2. 進階設定

- 調整圖片位置：透過按鈕切換，將圖片顯示在上下左右的方向。

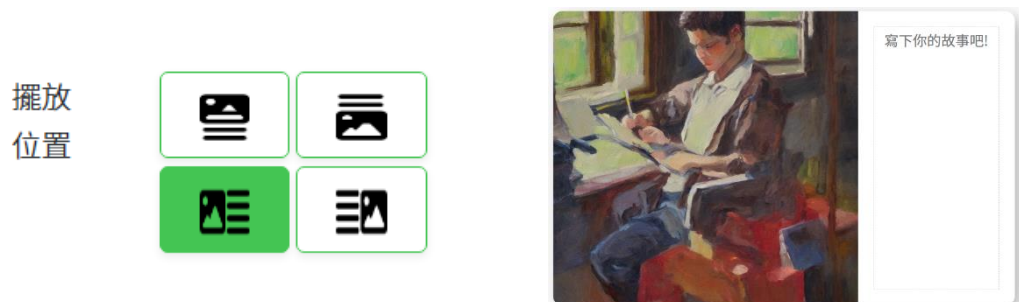


圖 14 調整圖片方向：左方

擺放
位置



圖 15 調整圖片方向：右方

參數設定：提供 Scale、Seed、Steps 及圖片長寬供使用者參數的調整，且預設值為選擇風格所選擇的參數設定。

3. 其他功能

- 圖片預覽區：將文字生成的圖片顯示在右方
- 文字旁白輸入：提供文字區塊，敘述生成圖像中的故事內容



圖 16 生成圖像預覽及旁白輸入區塊

4. 頁面選擇區塊

本系統提供頁面選擇的功能，可以新增、刪除並修改頁面，選擇指定的頁數，就可以進行編輯，也會保留前一頁的所有設定，以更順利完成繪本。

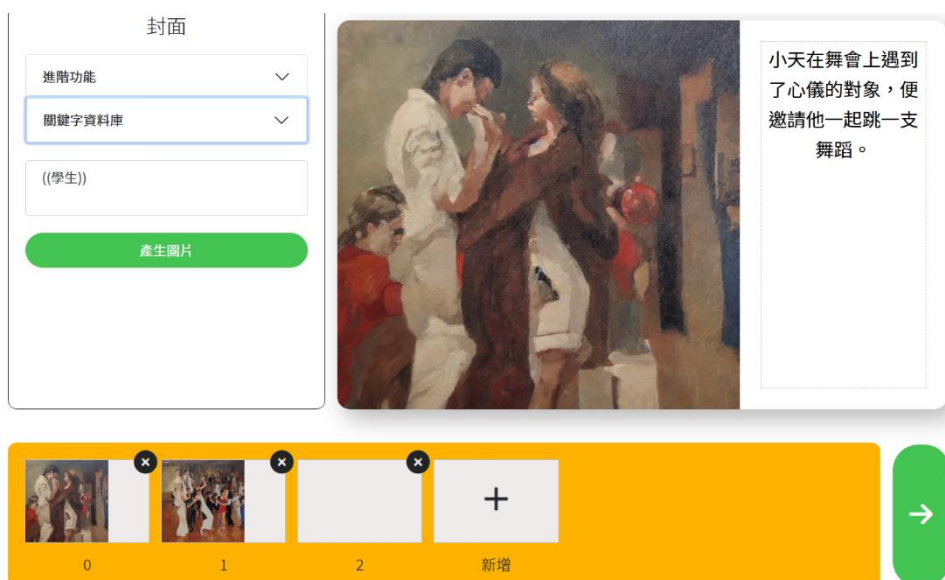


圖 17 頁面選擇區塊：封面頁

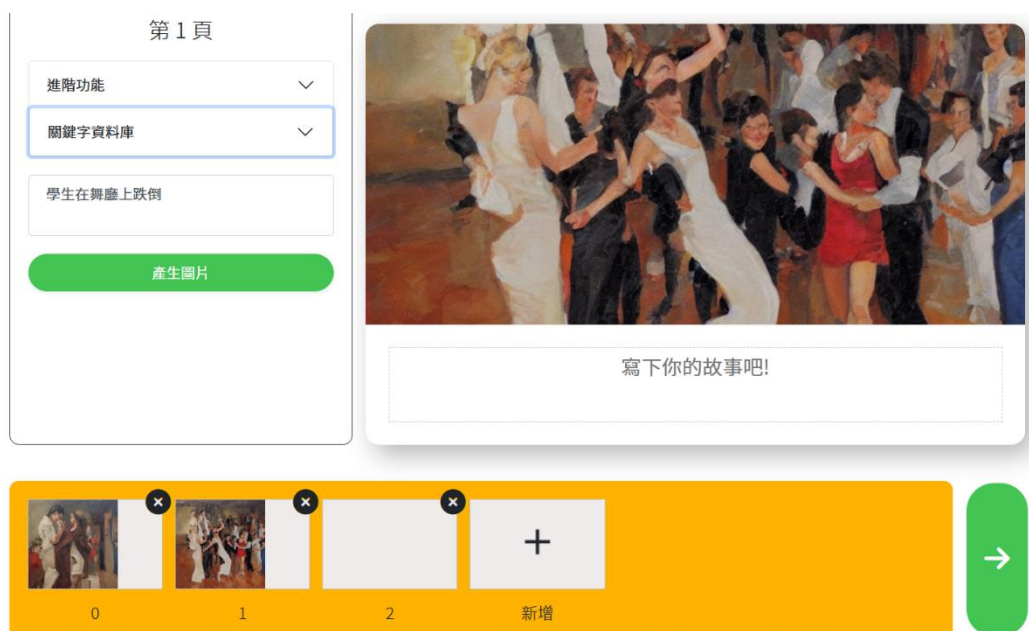


圖 18 頁面選擇區塊：第一頁

3. 儲存繪本：具有以下功能

- 選擇類別：選擇故事類別
- 狀態(公開/私人)：設定創作的繪本是否要公開在閱覽頁面
- 標題與大綱：創作故事的標題以及故事內容



圖 19 儲存頁面

(三) 會員機制

我們提供了會員機制，以方便使用者可以儲存自己的繪本以及提供後續創作，另外也方便其他使用者閱覽繪本時，可以知道繪本的作者資訊。



圖 20 登入頁面



圖 21 註冊頁面

1. 會員註冊及登入：提供會員以使用者名稱、電子郵件來建立會員。
2. 會員中心：紀載會員資料以及所創作的繪本

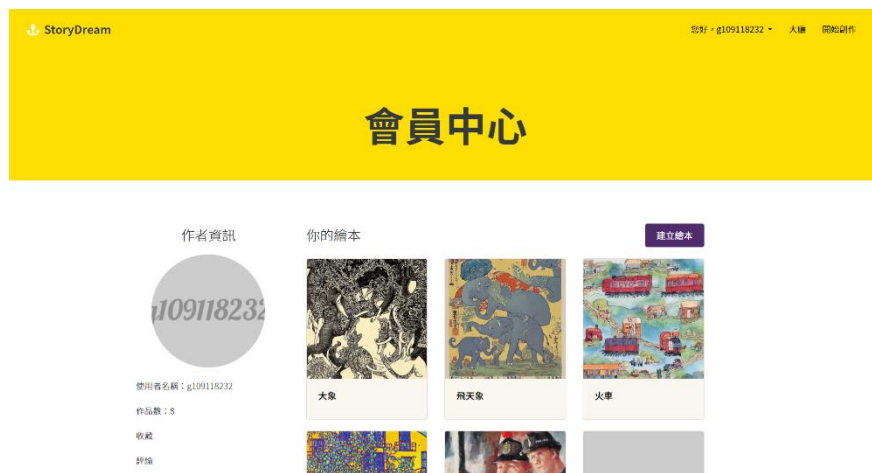


圖 22 會員中心：會員資料、繪本

二、研究貢獻

本計畫主要的創新與貢獻為：

1. 開發一個線上繪本創作平台「StoryDream 故事夢」，透過文字生成圖像技術提供中高年級學童創作電子繪本，也可在此平台中展示作品。
2. 不僅能作為學童培養寫作及閱讀的繪本創作平台，也可讓教師作為教學用途上的教具，學童在使用時也可以透過文字敘述的方式生圖，加強學童對於文字對圖像的連結記憶。

三、未來展望

由於開發時間是在 2022 年底，然而現今 2023 年下半年期間，在生圖的研究也迅速成長，如 Drag Your GAN 可以透過拖拉方式調整圖片、ControlNet 可以簡單透過畫骨架就能生成指定姿勢的人物圖像，這些都可以讓本系統的創作更便捷，且更具有挑戰性。

此外，由於本系統針對的客群是中高年級的學童，所以未來可以藉由文字轉語音的功能，提供敘述故事的便利性，也吸引學童們培養閱讀的良好習慣。

伍、參考文獻

- [1] 文本到圖像生成模型 (2023) 維基百科 檢自：<https://zh.wikipedia.org/zh-tw/%E6%96%87%E6%9C%AC%E5%88%B0%E5%9B%BE%E5%83%8F%E7%94%9F%E6%88%90%E6%A8%A1%E5%9E%8B>
- [2] Ajay Jain , Jonathan Ho , Pieter Abbeel (2020) Denoising Diffusion Probabilistic Models 。檢自：<https://arxiv.org/abs/2006.11239> (2022,1,15)
- [3] Stability AI Stable Diffusion launch announcement 檢自：<https://stability.ai/blog/stable-diffusion-announcement>
- [4] Björn Ommer , Dominik Lorenz , Patrick Esser , Robin Rombach (2022) High-Resolution Image Synthesis with Latent Diffusion ModelsStable diffusion 。檢自：<https://arxiv.org/abs/2112.10752> (2022,1,20)
- [5] getimg.ai — Guides to Stable Diffusion parameters 檢自：<https://getimg.ai/guides/stable-diffusion-parameters> (2023,1,23)
- [6] huggingface — Diffusers 檢自：<https://huggingface.co/docs/diffusers/index> (2022,12,15)
- [7] MDN (mozilla.org) — Django 介紹 檢自：<https://developer.mozilla.org/zh-TW/docs/Learn/Server-side/Django/Introduction>

陸、附件

一、本系統系統測試建置繪本



圖 23 連續性風格一致繪本製作範例 1



圖 25 連續性風格一致繪本製作範例 2



圖 24 連續性風格一致繪本製作範例 3

二、會員中心程式碼

檔案路徑	./account/views.py
功能敘述	載入處理會員中心頁面及功能
程式碼	
<pre>from django.shortcuts import render,redirect from django.http import HttpResponse from django.contrib.auth.forms import UserCreationForm from .forms import * from .models import Userinfo from MakerSpace.models import Book from django.views.decorators.csrf import csrf_exempt from django.contrib.auth import authenticate, login, logout from django.contrib.auth.decorators import login_required from django.contrib import messages from MakerSpace.models import * @csrf_exempt # Create your views here. def registerPage(request): form= CreateUserForm() if request.method =='POST': form = CreateUserForm(request.POST)</pre>	

```

if form.is_valid():

    form.save()

    user = form.cleaned_data.get('username')

    UserID = Userinfo.objects.create(UserID=user)

    messages.success(request,'Account was created for '+ user)

    return redirect('login')


context = { "form":form }

return render(request , 'account/register.html',context)


@csrf_exempt
def loginPage(request):

    if request.method == 'POST':

        uid= request.POST.get('username')

        password = request.POST.get('password')

        user = authenticate(request,username=uid, password=password)

        if user is not None:

            login(request,user)

            userID = Userinfo.objects.get(UserID=uid)

            rep = redirect(f'../{userID}/')

            rep.set_cookie("is_login", True)

            rep.set_cookie("uid", userID)

            return rep

```



```

        # return redirect(f'../?id={username}')

        #127.0.0.1:8000/account/

        #../ -> 127.0.0.1:8000/

        #f127.0.0.1:8000/?id={username}'

    else:

        messages.info(request,'Username or password is incorrect')

    return render(request , 'account/login.html')

def logoutUser(request):

    logout(request)

    return redirect('login')

# def style_choose(request

    # book = kwargs['book_id']

    # "<book_id>/"

    # {% csrf_token %}

    @login_required(login_url='login')

def accountCenter(request,*args,**kwargs):

    status = request.COOKIES.get('is_login')

    if status:

        userID = Userinfo.objects.get(UserID=kwargs['uid'])

        # request.POST.get('username')

```

```

book = Book.objects.filter(userinfo=userID)

context = {
    'UserID' : userID,
    'books' : book
}

# userID = request.POST.get('username')

# book = Book.objects.all()

# userinfo = Userinfo.objects.all()

# context = {'userinfo': userinfo,
#            'book_id':book}

#

rep = render(request , 'account/member.html', context)

# rep.set_cookie("is_login", True)

# rep.set_cookie("uid", userID)

return rep

return redirect('login')

```

三、閱覽大廳程式碼

檔案路徑	./storyhall/views.py
功能敘述	載入處理閱覽大廳顯示功能、顯示繪本功能及瀏覽功能
程式碼	
<pre> from django.shortcuts import render , redirect from django.shortcuts import redirect from MakerSpace.models import * from django.http import JsonResponse ,HttpResponse from django.views.decorators.csrf import csrf_exempt from django.contrib.auth.decorators import login_required </pre>	

```

from django import template

from django.template import loader ,Context

# Create your views here.

#####

# 顯示頁面 #

#####

@csrf_exempt
def showpage(request):

    bookID = Book.objects.get(id=int(request.POST.get('id')))

    #新頁面資訊顯示

    pages = Image.objects.get(book=bookID,page_number=request.POST.get('page'))

    setblock1 = loader.get_template('storyhall/showpage.html')

    c={

        'book':bookID,

        'pages':pages,

    }

    return HttpResponse(setblock1.render(c))

#####

# 顯示繪本 #

#####

def showStory(request , *args, **kwargs):

    #取得已經公開繪本的 ID

    book_id = int(kwargs['book_id'])

    book = Book.objects.get(id = book_id)

    pages = Image.objects.filter(book = book_id).order_by('page_number')

```

```

content = {
    'book' : book,
    'pages' :pages
}

return render(request,'storyhall/story_view.html' , content)

#####

# 瀏覽 #

#####

def index(request):
    #取得已經公開繪本的 ID、作者資訊
    # print(request.COOKIES.get("uid"))
    # print(request.COOKIES.get('is_login'))

    book = Book.objects.all()
    # book_cover = Image.objects.all()

    content = {
        'books':book,
    }

    return render(request,'storyhall/index.html',content)

    #若選擇後 post 接收風格的設定

    #取得 picturebookID、style prompt 後重新導向至繪本建立頁面

```

檔案路徑	./templates/storyhall/showpage.html
功能敘述	載入閱覽大廳顯示頁面
程式碼	
<pre> <div id="page-cover" {% if 'pages.img_location' == "area-top" %} class="area-top" {% else %} class="{{pages.img_location}}" {% endif %} style = "max-height:60vh;" > <div id="page-image" style="background-image: url('data:image/png;base64,{{ pages.image }}');" ></div> <div id="page-text"> <textarea id="ta-readonly" readonly>{{ pages.description }}</textarea> </div> </div> </pre>	

檔案路徑	./storyhall/templates/storyhall/story_view.html
功能敘述	載入閱覽大廳顯示繪本頁面
程式碼	
<pre>{% extends 'base.html' %}}{% block content%} <div class="container storyview"> <div class="row"> <div class="col-lg-3 col-sm-12"> <p class="text-center fz-24">作者資訊</p> <p class="text-center mt-10">作者名稱</p> <p class="text-center">{{ book.author }}</p> <p class="mt-10">故事名稱：{{ book.title }}</p> <p class="mt-10">大綱：{{ book.description }}</p> </div> <div class="col-1 d-none d-lg-block"> <div class="d-flex" style="height: 100%"> <div class="vr"></div> </div> </div> <div class="col-lg-8 col-sm-12"> <div class="row"></pre>	

```

<div class="col-12">{% include 'storyhall/showpage.html'%}</div>

<div class="col-12">

  <!-- 頁數選擇 -->

  <div

    class="container mt-30 of-y"

    style="padding-bottom: 3%; padding-top: 3%"

  >

    <div class="row">

      <ul class="ul-of">

        {% for page in pages %}

          <li

            class="photo"

            type="button"

            onclick="showpage({{ page.page_number }})"

            value="{{ page.page_number }}"

          >

            <img

              class="cover-img"

              id="page{{ page.page_number }}"

              style='background-image:url("data:image/png;base64,{{ page.image }}")'

            />

            <p class="page-num text-center">{{ page.page_number }}</p>

          </li>

        {% endfor %}

      </ul>

    </div>

  </div>

```

```

</div>

</div>

</div>

</div>

</div>

<!--顯示封面-->

<script>
    showpage(0);
</script>

<script>
    //初始化設定

    let init_page_location = document.getElementById("page-cover");
    let init_coverpage = document.getElementById("page-image");
    let init_coverpage_content = document.getElementById("ta-readonly");
    init_page_location.className = "{{pages.first.img_location}}";
    init_coverpage.style =
        "background-image: url('data:image/png;base64,{{pages.first.image}})";
    init_coverpage_content.value = "{{pages.first.description}}";

    //顯示所選頁面資訊

    function showpage(num) {
        $(document).ready(function () {
            $.ajaxSetup({
                data: { csrfmiddlewaretoken: "{{ csrf_token }}" },
            });
        });
    }

```



```
});  
  
$.ajax({  
    url: "../showpage/", // 要傳送的頁面  
    method: "POST", // 使用 POST 方法傳送請求  
    data: {  
        id: "{{ book.id }}",  
        page: num,  
    }, // 將表單資料用打包起來送出去  
    dataType: "html",  
    //若成功-> localStroge  
    success: function (response) {  
        $("#page-cover").replaceWith(response);  
    },  
});  
  
});  
  
}  
</script>  
  
{% endblock% }
```

檔案路徑	./templates/storyhall/index.html
功能敘述	載入閱覽大廳瀏覽頁面
程式碼	
<pre>{% extends 'base.html' %}}{% block content%} {% load static %} <div class="jumbotron jumbotron-lg jumbotron-fluid mb-0 pb-5 bg-warning position-relative"> <div class="container text-white h-100 tofront"> <div class="d-lg-flex align-items-center justify-content-between text-center pl-lg-5"> <div class="col pt-4 pb-4"> <h1 style="color: #343a3e; font-weight: 600" class="display-3">StoryDream</h1> <h4 style="color: #343a3e; font-weight: 400; margin-bottom: 2rem;" > 解開創作故事的夢境! </h4> <button type="button" class="btn btn-lg btn-outline-white btn-round" onclick="create()"> <h3 style="color: #343a3e; font-weight: 600"> 開始創作! </h3> </button> </div> <div class="col align-self-bottom align-items-right text-right h-max-380 h-xl-560 position-relative z-index-1"> </div> </div> </div> </div> <svg style="-webkit-transform: rotate(-180deg); -moz-transform: rotate(-180deg); -o- transform: rotate(-180deg); transform: rotate(-180deg); margin-top: -1px;" version="1.1"></pre>	

```

id="Layer_1"                                xmlns="http://www.w3.org/2000/svg"
xmlns:xlink="http://www.w3.org/1999/xlink" x="0px" y="0px" viewBox="0 0 1440
126" xml:space="preserve">

    <path                                class="bg-warning"                                d="M685.6,38.8C418.7-
11.1,170.2,9.9,0,30v96h1440V30C1252.7,52.2,1010,99.4,685.6,38.8z"></path>

</svg>

<!-- End Header -->

<div class="container mt-10" style="margin-bottom:30px;">

    <div class="row">

        <div class="col-12">

            <div class="btn-group">

                <button    type="button"    class="btn    btn-primary    dropdown-toggle"    data-
toggle="dropdown" aria-expanded="false">

                    所有類型

                </button>

                <div class="dropdown-menu">

                    <a class="dropdown-item" href="#">科幻</a>

                    <a class="dropdown-item" href="#">童話</a>

                    <a class="dropdown-item" href="#">所有類型</a>

                    <a class="dropdown-item" href="#">冒險</a>

                    <a class="dropdown-item" href="#">青少年文學</a>

                    <a class="dropdown-item" href="#">日記</a>

                </div>

            </div>

        </div>

        <div class="btn-group">

            <button    type="button"    class="btn    btn-primary    dropdown-toggle"    data-
toggle="dropdown" aria-expanded="false">

```

```

        最新上傳

</button>

<div class="dropdown-menu">

    <a class="dropdown-item" href="#">最新上傳</a>

    <a class="dropdown-item" href="#">熱門</a>

    <a class="dropdown-item" href="#">瀏覽次數最多</a>

</div>

</div>

    <button type="button" class="btn btn-primary d-none d-sm-block" style="float:
right" onclick="create()" >

        建立繪本

    </button>

</div>

</div>

</div>

</div>

<div class="container mt-10" style="margin-bottom:30px;">

<div class="row mt-10">

<div class="col-12">

<div class="row row-cols-1 row-cols-md-4 g-4">

    { % for book in books % } { % if book.public_status % }

<div class="col-lg-3 col-sm-1 col-xs-1">

    <a id="card-a" href="{ % url 'storyview' book.id % }">

        <div id="card-style" class="card h-100">

            { % if book.getCover != none % }

{% else %}



{% endif %}


<div class="card-body">

<div class="row">

<div class="col-6">

<h5 class="card-title">{{book.title}}</h5>

<h6>{{book.author}}</h6>

</div>

</div>

</div>

</div>

</a>

</div>

{%endif%} {% endfor %}

</div>

</div>

</div>

</div>

```

```
{% endblock% }
```

四、創作中心程式碼

檔案路徑	./MakerSpace/views.py
功能敘述	載入創作中心資料庫、選擇風格、新增刪除功能及生成圖片
程式碼	
<pre>from django.shortcuts import render, redirect import requests import torch from diffusers import DiffusionPipeline, EulerDiscreteScheduler from django.http import JsonResponse, HttpResponse from django.urls import resolve from deep_translator import GoogleTranslator from django.views.decorators.csrf import csrf_exempt from django.contrib.auth.decorators import login_required from django import template from django.template import loader, Context # 存檔案 from io import BytesIO import base64 from django.core.files.uploadedfile import InMemoryUploadedFile from PIL import Image from .models import * # 給隨機 seed</pre>	

```

import random

import os

import replicate

# google translate

# from google_trans_new import google_translator

# 類別資料庫及關鍵字資料庫導入

categories = Category.objects.all()

promptBase = PromptBase.objects.all().values()

SIZE = [size for size in range(128, 1025, 128)]


# - 取得 ID 跟頁碼(e.g. 3

def ChangeLocation(request):

    # - 取得 ID 跟頁碼(e.g. 3

    bookID = Book.objects.get(id=int(request.POST.get('id')))

    page = Image.objects.filter(

        book=bookID, page_number=request.POST.get('page'))

    page.update(img_location=request.POST.get('location'))

    return JsonResponse({'status': True})

#####

# 創作頁面##

# #####


@login_required(login_url='login')

@csrf_exempt

def makerspace(request, *args, **kwargs):

```

```

if request.method == "POST":

    bookID = Book.objects.get(

        id=int(request.POST['book_Id'])) # session['bookID']

    bookID.sid = request.POST["style_Id"]

    bookID.save()

    pages = Image.objects.filter(book=bookID).order_by('page_number')

    count = sum(1 for i in pages)

    if not (count):

        styleID = Stylebase.objects.get(

            styleID=request.POST['style_Id']) # 取得繪本 ID

        steps = styleID.steps

        Image.objects.create(page_number=0, book=bookID, seeds=random.randint(

            1, 4294967295), steps=steps, img_location="area-left")

    print(1)

    return redirect(f'{bookID.id}/')

print(0)

bookID = Book.objects.get(id=int(kwargs['book_id']))

pages = Image.objects.filter(book=bookID).order_by('page_number')

context = {

    'book': bookID,

    'pages': pages,

    'promptBase': promptBase,

    'categories': categories,

    'height': SIZE,

    'width': SIZE,

}

# 若無圖片先以第一頁為預設

```



```

print(2)

return render(request, 'gallery/makerspace.html', context)

#####

# 顯示頁面

# #####

@csrf_exempt
def showpage(request):

    bookID = Book.objects.get(id=int(request.POST.get('id'))) # 原頁面資訊儲存

    oldpage = Image.objects.filter(

        book=bookID, page_number=request.POST.get('old_page_num'))

    oldpage.update(description=request.POST.get('old_page_text'),

                    img_location=request.POST.get('old_page_cover'))

    # 新頁面資訊顯示

    pages = Image.objects.get(

        book=bookID, page_number=request.POST.get('page'))

    setblock1 = loader.get_template('makerspace/main.html')

    c = {

        'book': bookID,

        'pages': pages,

        'promptBase': promptBase,

        'categories': categories,

        'height': SIZE,

        'width': SIZE,

```

```

    }

    return HttpResponse(setblock1.render(c))

#####

# 移除功能

# #####

@csrf_exempt
def remove(request):

    # - 取得 ID 跟頁碼(e.g. 3

    bookID = Book.objects.get(id=int(request.POST.get('id')))

    count = len(Image.objects.filter(book=bookID))

    if count > 1:

        #- 將該頁面進行移除

        Image.objects.filter(

            book=bookID, page_number=request.POST.get('page')).delete()

        pages = Image.objects.filter(

            book=bookID).order_by('page_number') # 重新整理頁面

        #- 接著後面頁面的頁碼-1

        for page, item in enumerate(pages):

            item.page_number = page

            item.save()

        # item.update(page_number=int(page))

        pages = Image.objects.filter(book=bookID).order_by('page_number') # 重新整理頁面

        templates = loader.get_template('makerspace/loadpages.html')

        context = {'pages': pages, 'book': bookID}

```

```

return HttpResponseRedirect(templates.render(context))

#####

# 插入功能

# #####

@csrf_exempt
def insert(request):
    bookID = Book.objects.get(id=int(request.POST.get('id')))
    pages = Image.objects.filter(book=bookID).order_by('page_number')
    new_page = sum(1 for i in pages)
    seed = Image.objects.get(book=bookID, page_number=new_page-1).seeds
    Image.objects.create(page_number=new_page, book=bookID, seeds=seed,
                        steps=70, img_location="area-top",description="")
    pages = Image.objects.filter(book=bookID).order_by('page_number')
    # 重新設定幾個網頁
    templates = loader.get_template('makerspace/loadpages.html')
    context = {'pages': pages, 'book': bookID}
    return HttpResponseRedirect(templates.render(context))

def is_ajax(request):
    return request.META.get('HTTP_X_REQUESTED_WITH') == 'XMLHttpRequest'

#####

```

```

# 新版本

#####

# @login_required(login_url='login')

@csrf_exempt

def generate(request):

    #取得 bookID 所綁定的 style prompt

    print(request.POST['bid'])

    # translator = google_translator()

    bookID = Book.objects.get(id=request.POST['bid']) #session['bookID']

    pages = Image.objects.filter(book=bookID,page_number=request.POST['page'])

    if is_ajax(request=request) and request.method == "POST":

        #只要生成圖像參數表單即可

        prompt = GoogleTranslator(source='auto',
target='en').translate(request.POST['prompt'])#加上 style

        scale = float(request.POST['scale'])

        seed = int(request.POST['seed'])

        steps = int(request.POST['steps'])

        print(prompt,scale,seed,steps,sep='\n')

        """

        seed: 能控制圖片生成的多樣性，

        step: 次數越多，文本推理步驟就越多

        SCALE:

        """

        #Set the REPLICATE_API_TOKEN environment variable

        os.environ["REPLICATE_API_TOKEN"] =
        "r8_1g3WdOxVqzWy6IVIoXybqHf4BGzZuBj1IBrRF"

```

```

model = replicate.models.get("stability-ai/stable-diffusion")

version =
model.versions.get("db21e45d3f7023abc2a46ee38a23973f6dce16bb082a930b0c49861f96d1e5bf")

# https://replicate.com/stability-ai/stable-diffusion/versions/db21e45d3f7023abc2a46ee38a23973f6dce16bb082a930b0c49861f96d1e5bf#input

style = Stylebase.objects.get(styleID=str(bookID.sid))

inputs = {

    'prompt': prompt + style.stylePrompt ,

    'image_dimensions': "768x768",

    'num_outputs': 1,

    'num_inference_steps': steps,

    'guidance_scale': scale,

    'scheduler': "DPMSolverMultistep",

    'seed':seed,

}

# https://replicate.com/stability-ai/stable-diffusion/versions/db21e45d3f7023abc2a46ee38a23973f6dce16bb082a930b0c49861f96d1e5bf#output-schema

output = version.predict(**inputs)[0]

response = requests.get(output)

buffer = BytesIO(response.content)

#取得目前繪本 id 資料夾，若沒有則建立並依據編號存到該路徑中

image_str = base64.b64encode(buffer.getvalue()).decode("utf-8")

pages.update(prompt=request.POST['prompt'],image=image_str, height=768, width=768,seeds=seed,steps=steps, scale= scale)

return JsonResponse({'image_str': image_str})

```

```

#####

# 風格選擇頁面

# #####

# 建立繪本 ID-取得會員資料等

def style_choose(request, *args, **kwargs):

    # 取得繪本 ID

    book = kwargs['book_id']

    # 導入風格資料庫

    style = Stylebase.objects.all()

    context = {'stylebase': style, 'book_id': book}

    # 若選擇後 post 接收風格的設定

    # 取得 picturebookID、style prompt 後重新導向至繪本建立頁面

    # 將繪本 ID 傳送至頁面

    return render(request, 'stylebase/style_choose.html', context)

# 建立繪本 ID-取得會員資料等

@csrf_exempt

def book_create(request, *args, **kwargs):

    print(request.COOKIES.get("uid"))

    print(request.COOKIES.get('is_login'))

    if request.COOKIES.get('is_login'):

        userid = Userinfo.objects.get(UserID=request.COOKIES.get("uid"))

```

```

book = Book.objects.create(author=userid, userinfo=userid,title="",description=")

# 將繪本 ID 傳送至頁面

return HttpResponseRedirect(f'makerspace/style_choose/{book.id}/')

return HttpResponseRedirect('account/login')

# else:

#   uid = Userinfo.objects.get(UserID='guest')

```

檔案路徑	./MakerSpace/stylebase/style_choose.html
功能敘述	載入創作中心選擇風格頁面
程式碼	
<pre> {% extends 'base.html'% } {% block content% } <div class="jumbotron jumbotron-lg jumbotron-fluid mb-0 pb-5 bg-warning position- relative"> <div class="container text-white h-100 tofront"> <div class="row align-items-center justify-content-center text-center"> <div class="col-md-10"> <h1 class="display-3" style="color: #343a3e; font- weight: 600">選擇創作風格</h1> <hr> <h4 style="color: #343a3e; font-weight: 600"> 你可以從下列選單中，選擇接下來創作中所使用的風格，你的創作將會採取 指定風格。 </h4> </div> </div> </div> </div> </pre>	

```

<div class="container mb-10" style="margin-top:5%;">

  <div class="row">

    <div class="col-12">

      <form action="{ % url 'MakerSpace' % }" method="post">

        <input type="hidden" id="book_Id" name="book_Id" value="{{ book_id }}" />

        <div class="row">

          { % for style in stylebase % }

          <div class="col-sm-3 mt-10">

            <button name="style_Id" value="{{ style.styleID }}" class="btn btn-style-choose"
type="submit">

              <a
                  id="card-a"                  href="#style?ID={{ style.styleID }}"
onclick="getStyle('{{ style.styleID }})' ">

                <div id="card-style" class="card h-100">

                  <div class="card-body">

                    <h5 id="{{ style.styleID }}" class="card-title">

                      {{ style.name }}

                    </h5>

                  </div>

                </div>

              </a>

            </button>

          </div>

          { %endfor% }

        </div>

      </div>

    </form>

```



```

</div>

<script>

function getStyle(style_Id) {

    document.getElementById("style_Id").value = style_Id;

}

</script>

{% endblock%}

```

檔案路徑	./MakerSpace/gallery/makerspace.html
功能敘述	載入創作中心頁面
程式碼	
<pre> {% extends 'base.html' %} {% block content %} <div id="main">{% include 'makerspace/main.html'%}</div> <div id="loadpages" class="container mt-10 of-y"> {% include 'makerspace/loadpages.html' %} </div> <!--初始化設定--> <script> let bid = Number(document.getElementById("bid").value); let init_page = document.getElementById("page"); init_page.value = 0; </pre>	

```

let init_page_text = document.getElementById("text-area");

init_page_text.text = "{{ pages.first.description }}";

init_page_text.value = "{{ pages.first.description }}";

let init_page_location = document.getElementById("page-cover");

init_page_location.className = "{{ pages.first.img_location }}";

showpage(init_page.value);

</script>

```

<!-- 關鍵字資料庫部分 -->

```

<script type="text/javascript">

    let prompt = [];

    function changeColor(target) {

        let promptArea = document.querySelector("#promptArea");

        let keyword = "";

        let index = prompt.indexOf(target.value);

        if (target.id==0 && index == -1) {

            target.id = 1;

            prompt.push(target.value);

            keyword = target.value;

            target.style.backgroundColor = "yellow";

        } else if (target.id == 1 ){

            target.id = 2;

            keyword = "(" + prompt[index] + ")";

            prompt[index] = keyword;

            target.value = keyword;

            target.style.backgroundColor = "orange";

        } else if (target.id == 2) {

            target.id = 3;

```

```

keyword = "(" + prompt[index] + ")";

target.value = keyword;

prompt[index] = keyword;

target.style.backgroundColor = "red";

} else if (target.id == 3) {

target.style.backgroundColor = "";

prompt.splice(index, 1);

target.value = target.value.replace("(","").replace(")","");

target.id = 0;

}

promptArea.value = prompt.join(',');

// console.log(prompt+target.id+index);

}

function updatePrompt(){

txt = promptArea.value

prompt = txt.split(',');

prompt = prompt.filter(el => el);

promptArea.value = prompt.join(',');

}

function clear(items) {

let len = items.children.length;

for (let i = 0; i < len; i++) {

items.children[0].remove();

}

return items

```

```

}

function reloadPrompt(items,prompt){

  let promptbase = items.children;

  //console.log(prompt);

  //console.log(items.children);

  for(let i = 0 ; i < prompt.length;i++){ // prompt 資料庫

    let text = prompt[i];

    console.log(prompt[i])

    for(let j = 0 ; j < promptbase.length; j++ ){ // promptbase

      text = text.replace('(',").replace(')',").replace('(',").replace(')',")

      if(text == promptbase[j].innerHTML){

        if(prompt[i].includes("(")){

          promptbase[j].id = 3;

          promptbase[j].style.backgroundColor = "red";

        }else if(prompt[i].includes("")){

          promptbase[j].id = 2;

          promptbase[j].style.backgroundColor = "orange";

        }else{

          promptbase[j].id = 1;

          promptbase[j].style.backgroundColor = "yellow";

        }

        promptbase[j].value = prompt[i];

      }

    }

  }
}

```

```

    }

}

function showKeyword(id) {

    let pb = document.querySelector("#promptbase");


    //清空目前種類

    clear(pb);

    //判斷清單類別給予內容

    {% for prompt in promptBase %}

    //    console.log(id.value == "{{prompt.category_id}}");

    if(id.value == "{{prompt.category_id}}"){

        document.querySelector("#promptbase").innerHTML += '<button class="btn btn-outline-dark radius" style="margin: 0.2rem; padding: 0.1rem; width: 80px" type="button" id=0 onclick="changeColor(this)" value={{prompt.keyword}}>{{prompt.keyword}}</button>';

    }

    {%endfor%}

    reloadPrompt(pb,prompt);


    //返回種類清單

    let Return = document.createElement("button");

    Return.classList.add("btn");

    Return.classList.add("btn-outline-dark");

    Return.classList.add("radius");


    Return.style.margin = "0.2rem";

    Return.style.padding = "0.1rem";

```

```

Return.style.width = "80px";

Return.innerText = "  ";

Return.addEventListener("click", (e) => {

    document.querySelector("#promptbase").innerHTML = '{% for category in categories
% }<button      class="btn      btn-outline-dark      radius"      type="button"
onclick="showKeyword(this)"      style="margin: 0.2rem; padding: 0.1rem; width: 80px"
value={{ category.id }}>{{ category.name }}</button>{% endfor %}'

});

pb.appendChild(Return);

}

</script>

{% endblock %}

```

檔案路徑	./templates/makerspace/main.html
功能敘述	載入創作中心新增頁面
程式碼	
<pre> <div class="container markerspaceMain"> <div class="row"> <div class="col-lg-4 col-sm-12" style="padding: 20px; border-radius: 10px; border: 1px solid #333333;"> <form id="generate" method="POST" class="form-group needs-validation" novalidate> {% csrf_token %} <input type="hidden" id="bid" name="bid" value="{{ book.id }}" /> {% if pages.page_number == 0 %} </pre>	

```

<p id="show_page_number" class="text-center fz-24">封面</p>

{% else %}

<p id="show_page_number" class="text-center fz-24">第 {{pages.page_number}}
頁</p>

{% endif %}

<input id="page" type="hidden" name="page" value="{{pages.page_number}}"
readonly="readonly" />

{% include 'makerspace/left.html'%}

</form>

</div>

<div class="col-lg-8 col-sm-12 makerspaceImg">

{% include 'makerspace/right.html'%} {% include 'generate.html'%}

</div>

</div>

</div>

```

檔案路徑	./templates/makerspace/loadpages.html
功能敘述	載入創作中心刪除頁面
程式碼	
<pre> <!-- 頁數選擇 --> <div class="container mt-10 makerspaceLoadpages"> <div class="row"> <div id="scrollpage" class="col-lg-11 col-sm-12 loadpages of-y white-block" </pre>	

```

style="padding-bottom: 0px"

>
<ul class="ul-of" style="padding-left: 0">
    { % for page in pages % }
    <li
        class="photo"
        type="button"
        onclick="showpage({ { page.page_number } })"
        value="{ { page.page_number } }"
    >
        <img
            class="cover-img"
            id="page{ { page.page_number } }"
            style="background-size: contain;background-repeat: no-repeat;background-
image:url('data:image/png;base64,{ { page.image } }')"
        />
        <div
            class="image-remove"
            id="del"
            type="button"
            onclick="remove({ { page.page_number } })"
        >
            <i class="fa-solid fa-circle-xmark"></i>
        </div>
        <p class="page-num text-center">{ { page.page_number } }</p>
    </li>

    { % endfor % }

```



```

<li id="add" class="photo text-center" type="button">

  <img

    style="

      background-image:
url('data:image/png;base64,iVBORw0KGgoAAAANSUheUgAAAJUAAABkCAIAAA
AADuWkAAAAAXNSR0IArs4c6QAAAAARnQU1BAACxjwv8YQUAAAAJcEhZcwA
AFiUAABYIAUISJPAAAAIASURBVHhe7di9SsNQHEDxvoqDg4Pg4ObgF/okIqWKF
FwV30Kd1UmfQBf7DkUdikPBQUw/tLRFaEWDKUWuUCsU7UIP+A3hf4dATi4JyTx
Xq7Uo+u6lXm+3Wt1OJ5hrTLx/Hplg2mw0Xtvt243mP+j+BlqN5vPtVown2RJvPjo9at
HUfXpqb88KtVvE41WuP8EUKn0z0fQL967iWCu3+rdxi95fjQB+69SSfHjhe2X6irD8/
3HZj82+7HZj81+bPZjsx+b/djsx2Y/Nvux2Y/Nfmz2Y7Mfm/3Y7MdmPzb7sdmPzX5s9
mOzH5v92OzHZj+2NPe7urw6PTmLnZ9fPJTLwWo6pLlfbmtnanomtri8dlMsBqvpYD8
2+7HZj81+bPZjsx+b/djsx2Y/NvuxUfsdHR7n87uDL2uJ/1m5+azue1gNbC3f3BfKgVX
GX/Ufv29NSrQPWq/Hvv9KfslqP3id1V8uwfb2MwmbRYWVwrXhWA1cHd7Gz0+Blc
Zf9R+w/D7k81+bPZjsx+b/djsx2Y/Nvux2Y/Nfmz2Y+v/I4X+2xxGmvtNAvux2Y/Nfmz
2Y7Mfm/3Y7MdmPzb7sdmPzX5s9mOzH5v92OzHZj82+7HZj81+bPZjsx+b/djsx2Y/N
vux2Y/Nfmz2Y7Mfm/3Y7MdmPzb7sdmPzX5s9mOzH1kUfQAfkkmdnfdJ2AAAAABJ
RU5ErkJggg==');

    "

  />

  <p class="page-num">新增</p>

</li>

</ul>

</div>

<div class="col-lg-1 col-sm-12">

  <form action=" .././savebook/{ {book.id} }/" method="post">

    <div class="d-grid gap-2 markerspaceBtn">

      <button

        class="btn btn-success btn-round btn-block shadow-sm"

        type="submit"

        onclick="showpage({ {pages.page_number} });"

      >

        <i class="fa-solid fa-arrow-right fa-2x"></i>

      </button>

```

```

</div>

</form>

</div>

</div>

</div>

{% include 'loadpagefunction.html' %}

```

檔案路徑	./templates/generate.html
功能敘述	載入創作中心生成圖片頁面
程式碼	
<pre> <!--生圖功能--> <script> \$(document).ready(function () { \$.ajaxSetup({ data: { csrfmiddlewaretoken: "{{ csrf_token }}" }, }); \$("#generate").on("submit", function (e) { e.preventDefault(); \$("#page-image").html("<div>開始生圖</div>"); \$.ajax({ url: "../generate/", // 要傳送的頁面 method: "POST", // 使用 POST 方法傳送請求 dataType: "json", // 資料格式 data: \$("form").serialize(), // 將表單資料用打包起來送出去 </pre>	

```

beforeSend: function () {

    // setting a timeout

    $("#page-image").html(

        '<div      class="spinner-border"      role="status"><span      class="visually-
hidden">Loading...</span></div role="status"><div>開始圖中... 請稍後</div>'

    );

    $("#page-image").css(

        "background-image",

        `url("data:image/png;base64,None")`

    );

},

success: function (response) {

    let image = response.image_str;

    $("#page-image").html("");

    $("#page-image").css(

        "background-image",

        `url("data:image/png;base64,${image}")`

    );

    $(`#page${$("#page").val()}`).css(

        "background-image",

        `url("data:image/png;base64,${image}")`

    );

},

error: function () {

    $("#page-image").html(`出現錯誤，請重新生圖`);

},

});

return false; // 阻止瀏覽器跳轉到"generate/"，因為已經用 ajax 送出去了

```

```
});

});

</script>
```

五、儲存頁面程式碼

檔案路徑	./savebook/views.py
功能敘述	載入儲存功能
程式碼	
<pre>from django.shortcuts import render from django.http import JsonResponse ,HttpResponse from django.shortcuts import render, redirect from django.views.decorators.csrf import csrf_exempt from MakerSpace.models import * # Create your views here. @csrf_exempt def save(request , *args, **kwargs): if request.method == "POST": if request.POST['public_status'] == 'private': status = False else: status = True book_id = int(kwargs['book_id']) book = Book.objects.filter(id = book_id) book.update(title=request.POST['book_title'],description=request.POST['description'],public_status=status)</pre>	

```

    #,cover_page=request.POST['cover_page']

    ID = Book.objects.get(id = book_id).author

    return redirect(f'../../?id={ID}')

@csrf_exempt
def savebook(request,*args, **kwargs):
    book_id = int(kwargs['book_id'])
    book = Book.objects.get(id = book_id)
    cover_page = Image.objects.get(book=book_id , page_number = 0)
    pages = len(Image.objects.filter(book=book_id))
    settings={
        'book':book,
        'cover_page':cover_page,
        'pages':pages
    }
    return render(request,'savebook/save.html',settings)

```

檔案路徑	./savebook/templates/savebook/save.html
功能敘述	載入儲存頁面
程式碼	
<pre> {% extends 'base.html' %} {% block content %} <div class="jumbotron jumbotron-lg jumbotron-fluid mb-0 pb-5 bg-warning position- relative"> <div class="container text-white h-100 tofront"> <div class="row align-items-center justify-content-center text-center"> </pre>	

```

<div class="col-md-10">
    <h1 class="display-3" style="color: #343a3e; font-
weight: 600">儲存繪本</h1>

    <hr>

    <h4 style="color: #343a3e; font-weight: 600">
        恭喜你即將完成繪本囉！填寫下方資料儲存繪本。
    </h4>

</div>

</div>

</div>

</div>

<form action="{ % url 'save' book.id % }" method="post">

<div class="container mt-10 mb-10" style="margin-top:5%;">

<div class="row">

<!--儲存設定-->

<div class="col-lg-6 col-sm-12">

<!--狀態列-->

    { % if cover_page.image != none % }

    { % else % }

{% endif %}

</div>

<!--儲存設定-->

<!--作者資訊-->

<div class="col-lg-6 col-sm-12">

<div class="mb-3">

<label class="form-label">類別：</label>

<select class="form-select" aria-label="Default select example">

<option selected disabled>請選擇類別</option>

<option value="1">類別 1</option>

<option value="2">類別 2</option>

<option value="3">類別 3</option>

</select>

<div class="invalid-feedback">請選擇類別</div>

</div>

<p>頁數：{{pages}}頁</p>

<label class="form-label">狀態：</label>

<div class="form-check form-check-inline">

<input

class="form-check-input"

type="radio"

name="public_status"

```

```

        id="open"

        value="public"

    />

    <label class="form-check-label" for="open"> 公開 </label>

</div>

<div class="form-check form-check-inline">

    <input

        class="form-check-input"

        type="radio"

        name="public_status"

        id="private"

        value="private"

        checked

    />

    <label class="form-check-label" for="private"> 私密 </label>

</div>


<div class="mb-3">

    <label for="book_title" class="form-label">故事標題： </label>

    <input

        type="text"

        class="form-control"

        id="book_title"

        name="book_title"

        placeholder="輸入故事標題"

        value="{{ book.title }}"

```



```

/>

<div class="invalid-feedback">請輸入故事標題</div>

</div>

<div class="mb-3">

  <label for="description" class="form-label">大綱：</label>

  <textarea

    class="form-control"

    id="description"

    name="description"

    placeholder="輸入故事大綱"

    rows="10"

  >{{book.description}}</textarea>

  <div class="invalid-feedback">請輸入大綱</div>

  <div class="d-grid gap-2">

    <button class="btn btn-success btn-round btn-block shadow-sm mt-10"
type="submit">儲存</button>

  </div>

</div>

<!--作者資訊-->

</div>

</div>

</form>

{% endblock%}

```