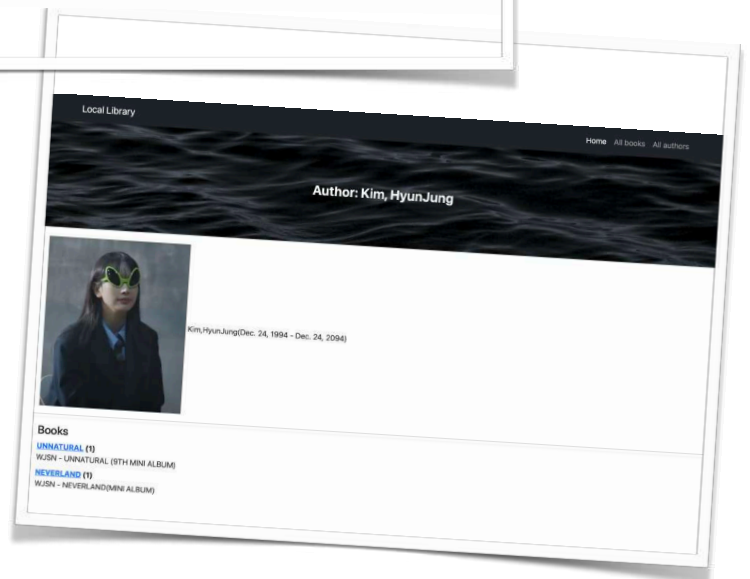
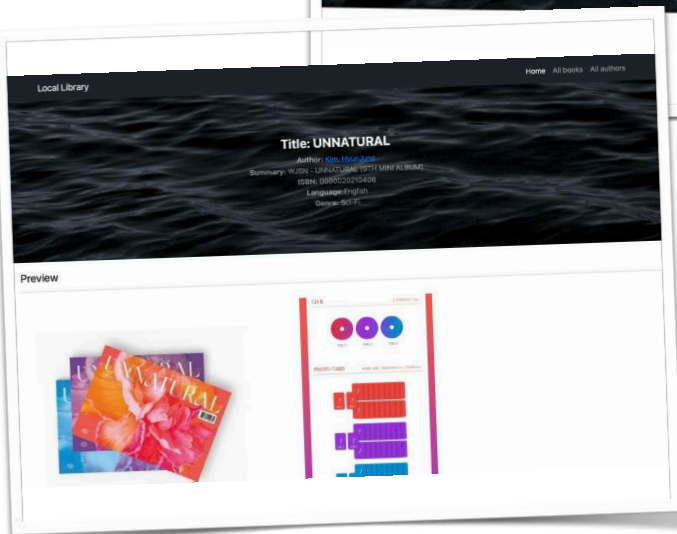
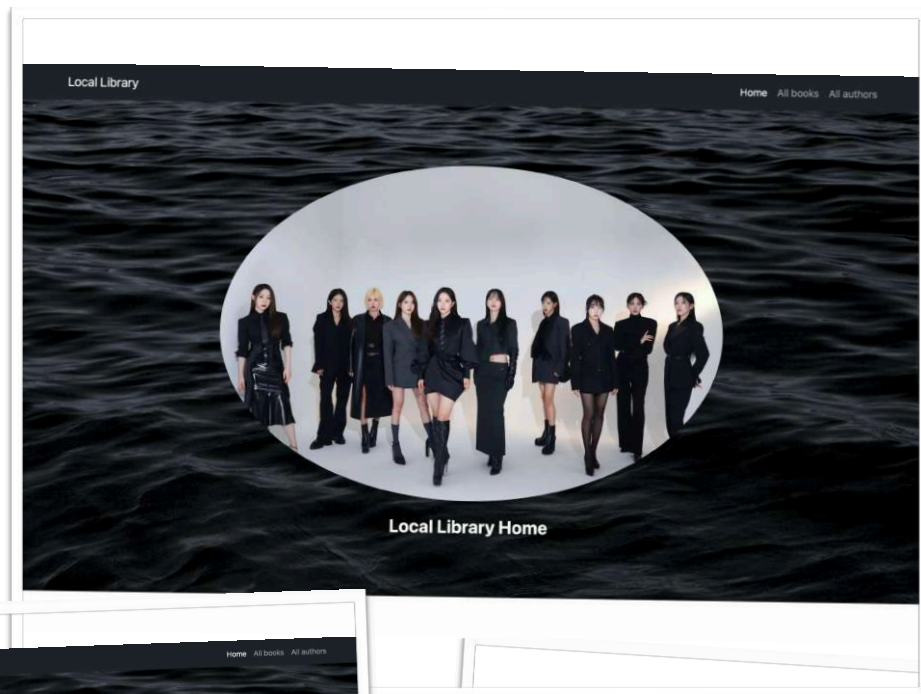


# 111-1 OOP

## Final Project

### Library management system



# Django簡介

## Django 的設計模式-MTV

- **Model** : 定義一些資料庫的東西 ( ORM ) , 通常直接和資料有關。
- **Template** : 使用者基本上就是看到這層, 也就是最後所呈現的 Template ( html ) 。
- **View** : 可以將這層看做是中間層, 主要負責 Model 和 Template 之間的業務邏輯。

## MTV的優點

- 可以明確且很快速的知道問題在哪裡, 並找到對應層去修改

## 建立開發環境

- 建立虛擬環境--myworld

指令：

```
mkdir django(創建django資料夾)
```

```
cd django(進入django資料夾)
```

```
mkdir projects(創建projects資料夾)
```

```
cd projects(進入projects資料夾)
```

```
python -m venv myworld(創建虛擬環境myworld)
```

```
source myworld/bin/activate(activate 虛擬環境)
```

- 在該環境中安裝Django

指令：

```
pip install Django
```

## 建立 Django Project

- 建立Project--locallibrary

指令：

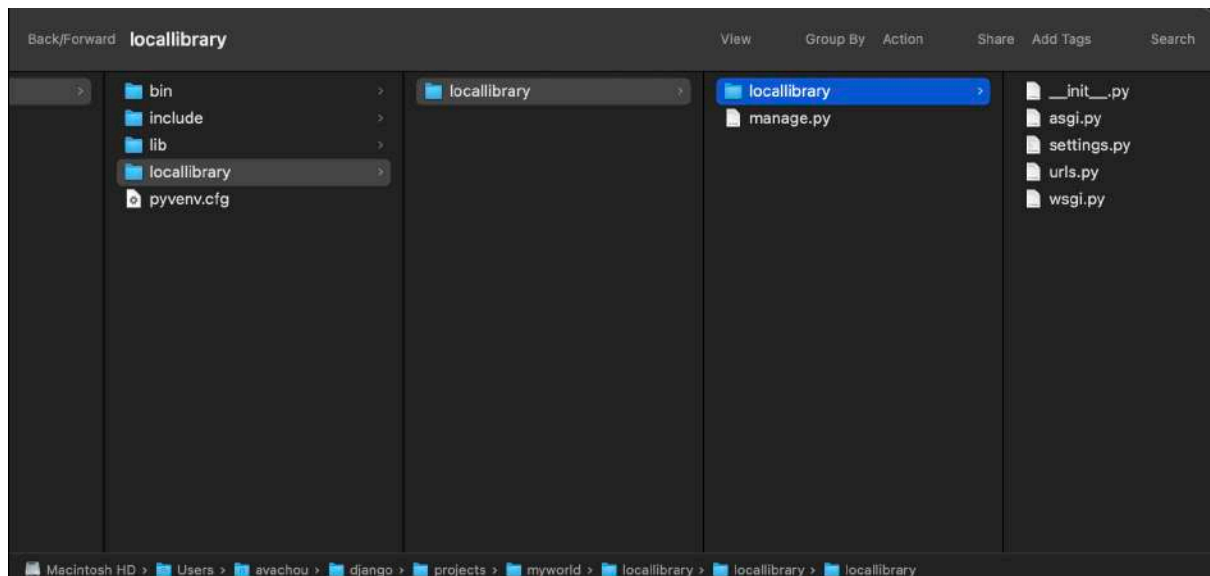
```
cd myworld
```

```
mkdir locallibrary
```

```
cd locallibrary
```

```
django-admin startproject locallibrary
```

```
cd locallibrary
```



- `__init__.py` 是一個空文件，指示 Python 將此目錄視為 Python 套件。
- `settings.py` 包含所有的網站設置。這是可以註冊所有創建的應用的地方，也是靜態文件，數據庫配置的地方。

- **urls.py**定義了網站 url 到 view 的映射。雖然這裡可以包含所有的 url，但是更常見的做法是把應用相關的 url 包含在相關應用中。
- **wsgi.py** 幫助 Django 應用和網絡服務器間的通訊。可以把這個當作模板。
- **manage.py**腳本可以創建應用，和資料庫通訊，啟動開發用網絡服務器。

### 啟動網頁伺服器進行測試

- 執行Project

指令：

```
python manage.py runserver
```

\*manage.py 是 Django 提供管理專案用的命令列工具

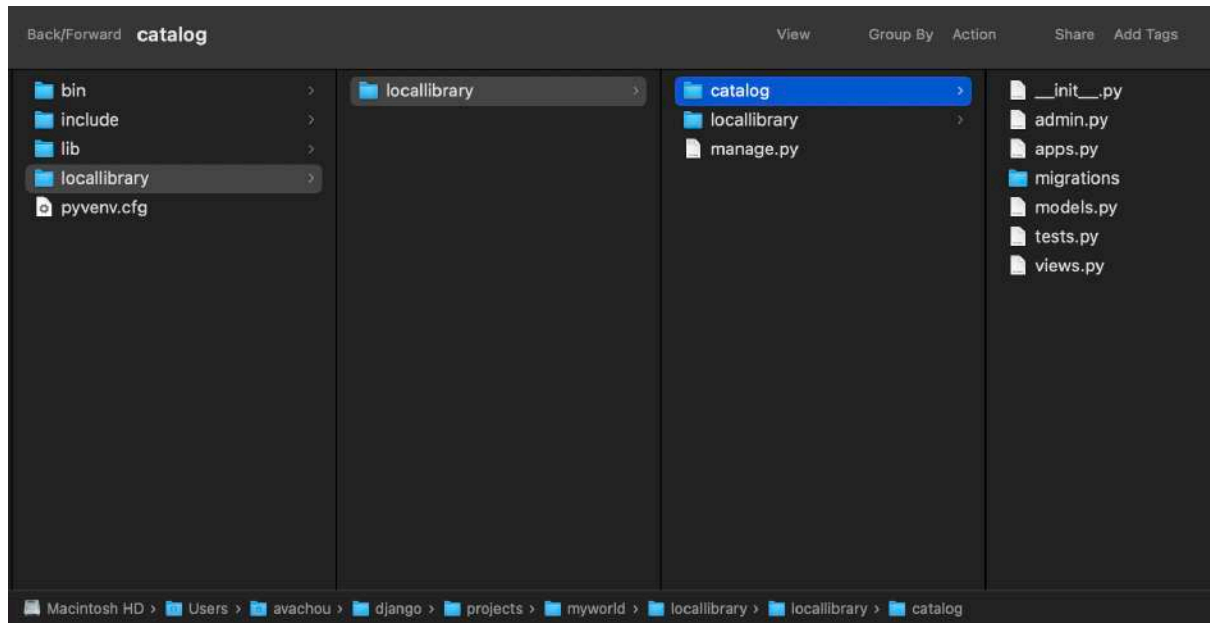
### 建立應用程式 **app**

可視為本專案建立的套件，通常會有多個 app 進行搭配，負責不同的功能

- 新增APP--catalog

指令：

```
python manage.py startapp catalog
```



- migrations，當修改數據模型時，這個文件會自動升級資料庫。
- \_\_init\_\_.py –一個空文件，Django/Python 會將這個文件作為Python 套件包並允許在項目的其他部分使用它。
- views.py，視圖函數
- models.py，模型
- tests.py，測試
- admin.py，網站管理設置
- apps.py，註冊應用

### 建立**Project**所需之靜態檔案目錄

- 建立templates,static等資料夾

指令：

```
mkdir templates
```

```
mkdir static
```

```
mkdir static/images
```

```
mkdir static/css
```

```
mkdir static/js
```

### 編輯專案環境設定檔--**locallibrary/settings.py**

- 設定應用程式

```
# Application definition

INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'catalog.apps.CatalogConfig',
]
```

- 設定樣板目錄的路徑

```
TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [BASE_DIR / "templates"],
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [
                'django.template.context_processors.debug',
                'django.template.context_processors.request',
                'django.contrib.auth.context_processors.auth',
                'django.contrib.messages.context_processors.messages',
            ],
        },
    ],
]
```

- 設定靜態檔案的路徑

```
STATIC_URL = '/static/'  
  
STATICFILES_DIRS = [  
    |     BASE_DIR / "static",  
    |  
]
```

- 配置資料庫

```
DATABASES = {  
    |     'default': {  
    |         |     'ENGINE': 'django.db.backends.sqlite3',  
    |         |     'NAME': BASE_DIR / 'db.sqlite3',  
    |         |  
    |         }  
    |  
}
```

## 網址的對應與委派

- 設定網址路由：locallibrary/urls.py

管理專案的套件目錄下的 urls.py，內有 urlpatterns 這個 list 負責網址與view(應用程式中函數)的對應

- 設定應用程式中函數的對應：catalog/urls.py

定義會被 urls.py import 的函數

## Django存取資料庫

- 建立內建APP所需之資料表

位於 locallibrary/settings.py 的內建 APP(admin / auth / contenttypes / sessions)，運作時所需之資料表(預設資料庫：sqlite3)，可透過以下指令產生

指令：

```
python manage.py migrate
```

- 異動資料模型 models.py

資料模型預設是以應用程式作為對應的單位，根據資料表定義，在 model.py 輸入程式碼，類別代表資料表，變數代表欄位定義

- 將模型的異動產生 migrations 檔

透過 makemigrations 命令，Django 會檢測到我們對資料模型的修改，並把這些修改的部分，儲存成一次 migration

指令：

```
python manage.py makemigrations
```

- 以 migrate 指令將遷移套用於資料庫

migrate 命令會將還沒執行過的遷移(migrations)，執行實際的資料庫異動

指令：

```
python manage.py migrate
```

### 定義 **LocalLibrary** 模型

catalog 中打開 models.py

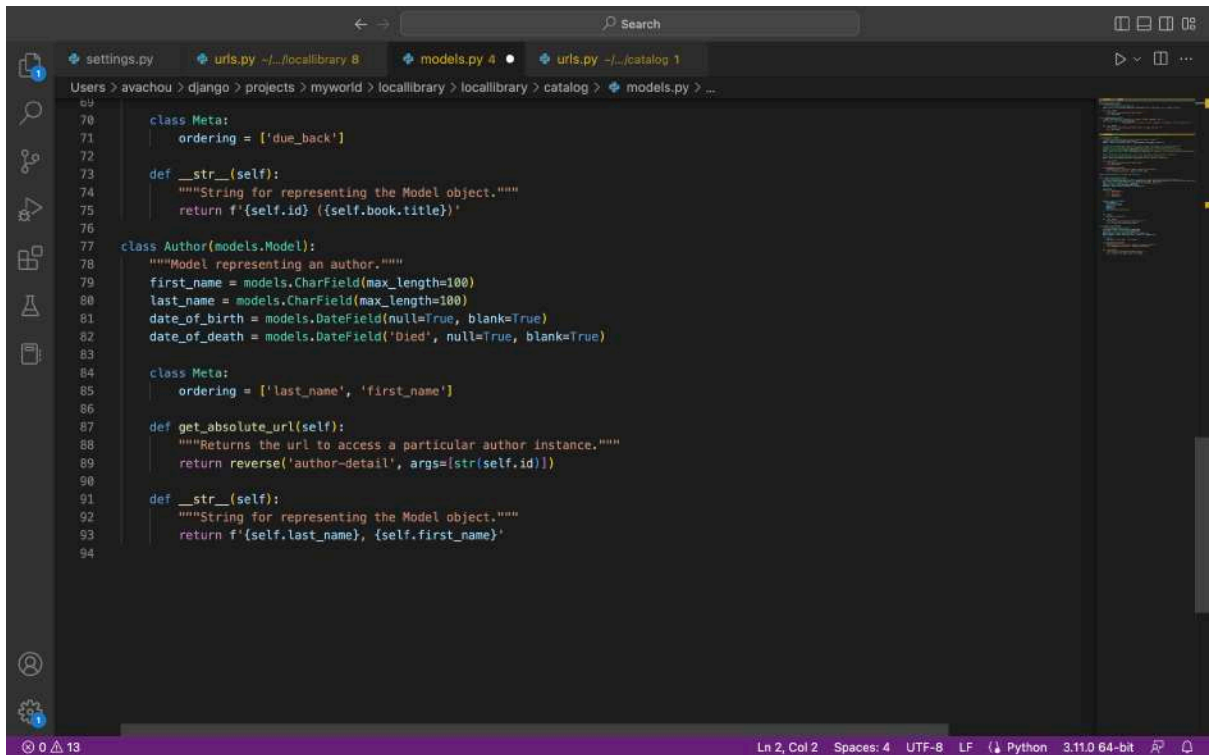
- 書籍類型模型 (Genre model): 儲存書籍類型的資訊
- 書本模型 (Book model): 一個可用書本的所有資訊
- 書本詳情模型 (BookInstance model): 一個特定的書籍副本



- 作者模型(Author model):作者資訊

```
Users > avachou > django > projects > myworld > locallibrary > locallibrary > catalog > models.py > ...
1 from django.db import models
2
3 # Create your models here.
4 class Genre(models.Model):
5     """Model representing a book genre."""
6     name = models.CharField(max_length=200, help_text='Enter a book genre (e.g. Science Fiction)')
7
8     def __str__(self):
9         """String for representing the Model object."""
10        return self.name
11
12 class Language(models.Model):
13     """Model representing a Language (e.g. English, French, Japanese, etc.)"""
14     name = models.CharField(max_length=200,
15                             help_text='Enter the book's natural language (e.g. English, French, Japanese etc.)')
16
17     def __str__(self):
18         """String for representing the Model object (in Admin site etc.)"""
19        return self.name
20
21
22 from django.urls import reverse #Used to generate URLs by reversing the URL patterns
23
24 class Book(models.Model):
25     """Model representing a book (but not a specific copy of a book)."""
26     title = models.CharField(max_length=200)
27     author = models.ForeignKey('Author', on_delete=models.SET_NULL, null=True)
28
29     # Foreign Key used because book can only have one author, but authors can have multiple books
30     # Author as a string rather than object because it hasn't been declared yet in the file.
31     summary = models.TextField(max_length=1000, help_text='Enter a brief description of the book')
32     isbn = models.CharField('ISBN', max_length=13, help_text='13 Character <a href="https://www.isbn-international.org/content/what-is-ISBN">ISBN</a>')
33
34     # ManyToManyField used because genre can contain many books. Books can cover many genres.
35     # Genre class has already been defined so we can specify the object above.
```

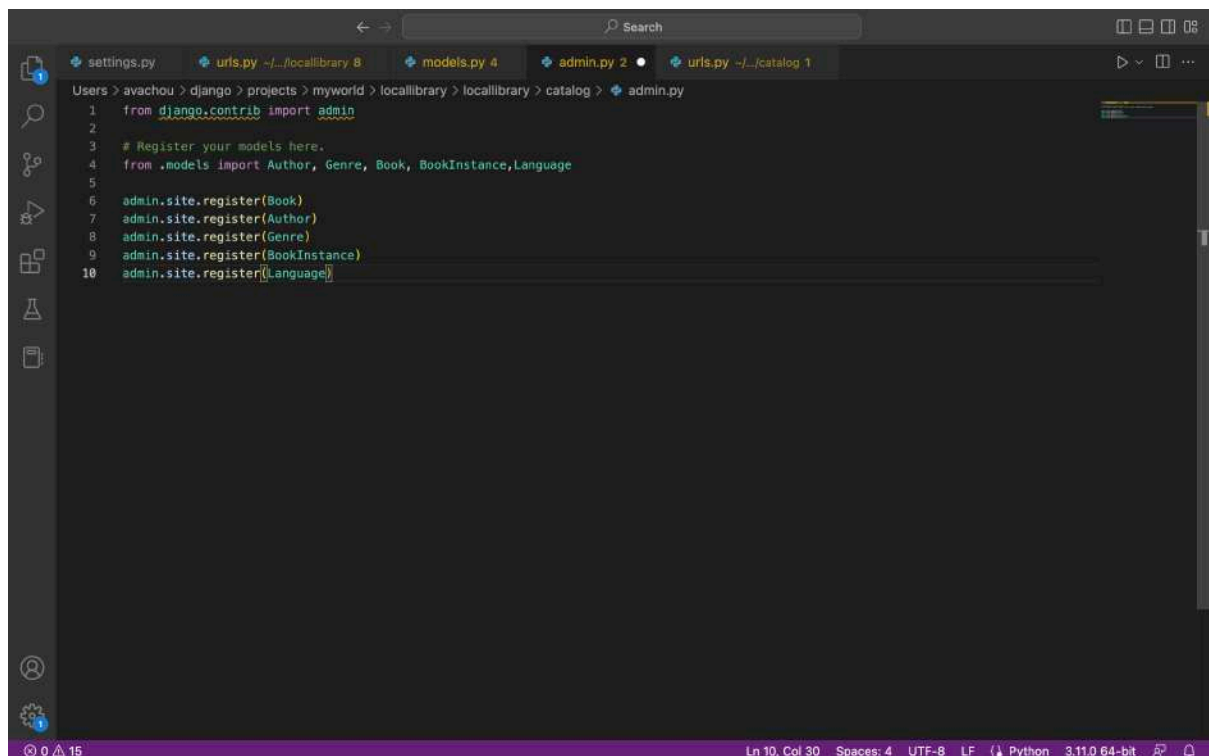
```
35 # Genre class has already been defined so we can specify the object above.
36 genre = models.ManyToManyField(Genre, help_text='Select a genre for this book')
37
38 def __str__(self):
39     """String for representing the Model object."""
40     return self.title
41
42 def get_absolute_url(self):
43     """Returns the url to access a detail record for this book."""
44     return reverse('book-detail', args=[str(self.id)])
45
46 import uuid # Required for unique book instances
47
48 class BookInstance(models.Model):
49     """Model representing a specific copy of a book (i.e. that can be borrowed from the library)."""
50     id = models.UUIDField(primary_key=True, default=uuid.uuid4, help_text='Unique ID for this particular book across whole library')
51     book = models.ForeignKey('Book', on_delete=models.SET_NULL, null=True)
52     imprint = models.CharField(max_length=200)
53     due_back = models.DateField(null=True, blank=True)
54
55     LOAN_STATUS = (
56         ('m', 'Maintenance'),
57         ('o', 'On loan'),
58         ('a', 'Available'),
59         ('r', 'Reserved'),
60     )
61
62     status = models.CharField(
63         max_length=1,
64         choices=LOAN_STATUS,
65         blank=True,
66         default='m',
67         help_text='Book availability',
68     )
69
```



```
69
70 class Meta:
71     ordering = ['due_back!']
72
73 def __str__(self):
74     """String for representing the Model object."""
75     return f'{self.id} {self.book.title}'
76
77 class Author(models.Model):
78     """Model representing an author."""
79     first_name = models.CharField(max_length=100)
80     last_name = models.CharField(max_length=100)
81     date_of_birth = models.DateField(null=True, blank=True)
82     date_of_death = models.DateField('Died', null=True, blank=True)
83
84     class Meta:
85         ordering = ['last_name', 'first_name']
86
87     def get_absolute_url(self):
88         """Returns the url to access a particular author instance."""
89         return reverse('author-detail', args=[str(self.id)])
90
91     def __str__(self):
92         """String for representing the Model object."""
93         return f'{self.last_name}, {self.first_name}'
94
```

## 註冊模型(Registering models )

從 catalog app 中打開 **admin.py**



```
1 from django.contrib import admin
2
3 # Register your models here.
4 from .models import Author, Genre, Book, BookInstance, Language
5
6 admin.site.register(Book)
7 admin.site.register(Author)
8 admin.site.register(Genre)
9 admin.site.register(BookInstance)
10 admin.site.register(Language)
```

## Admin 後台管理

Django 會自動建立管理的後台，具體來說就是透過 admin 應用程式管理 db.sqlite3 內的使用者、群組與對應模型的資料表

- 建立管理員帳號

指令：

```
python manage.py createsuperuser
```

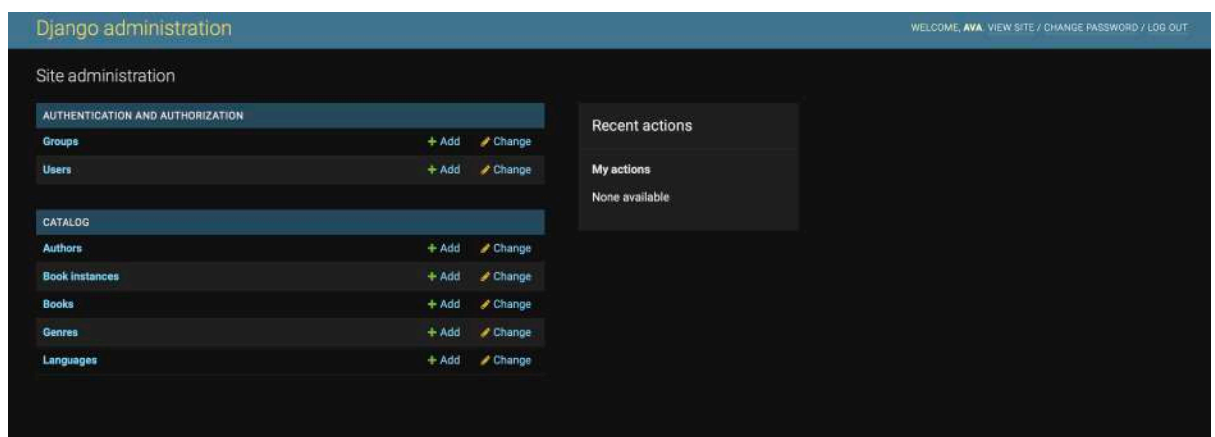
```
(myworld) (base) avachou@AvadeMacBook-Pro locallibrary % python manage.py createsuperuser
Username (leave blank to use 'avachou'): ava
Email address: 109207444@g.nccu.edu.tw
Password:
Password (again):
Superuser created successfully.
```

- 啟動伺服器並打開 admin 頁面

指令：

```
python manage.py runserver
```

接著輸入網址：<http://127.0.0.1:8000/admin/>



- 在頁面中新增書籍等等資訊

Django administration

WELCOME **AVA** / VIEW SITE / CHANGE PASSWORD / LOG OUT

Home / Catalog / Books / Add book

Start typing to filter...

AUTHENTICATION AND AUTHORIZATION

Groups + Add

Users + Add

CATALOG

Authors + Add

Book instances + Add

Books + Add

Genres + Add

Languages + Add

Add book

Title

Author

Summary

Enter a brief description of the book

ISBN

13 Character ISBN number

Genre

Science Fiction

Mystery

Hor

Select a genre for this book. Hold down "Control", or "Command" on a Mac, to select more than one.

Save and add another

Save and continue editing

SAVE

Django administration

WELCOME **AVA** / VIEW SITE / CHANGE PASSWORD / LOG OUT

Home / Catalog / Authors

Start typing to filter...

AUTHENTICATION AND AUTHORIZATION

Groups + Add

Users + Add

CATALOG

Authors + Add

Book instances + Add

Books + Add

Genres + Add

Languages + Add

Select author to change

ADD AUTHOR +

Action

Go

0 of 3 selected

☐

AUTHOR

☒

Kim, HyunJung

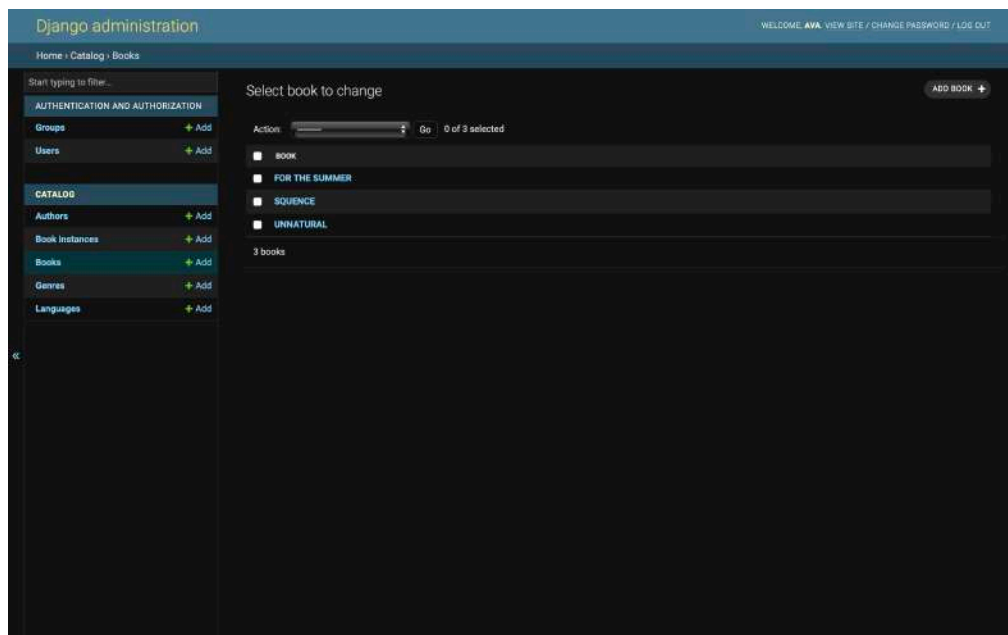
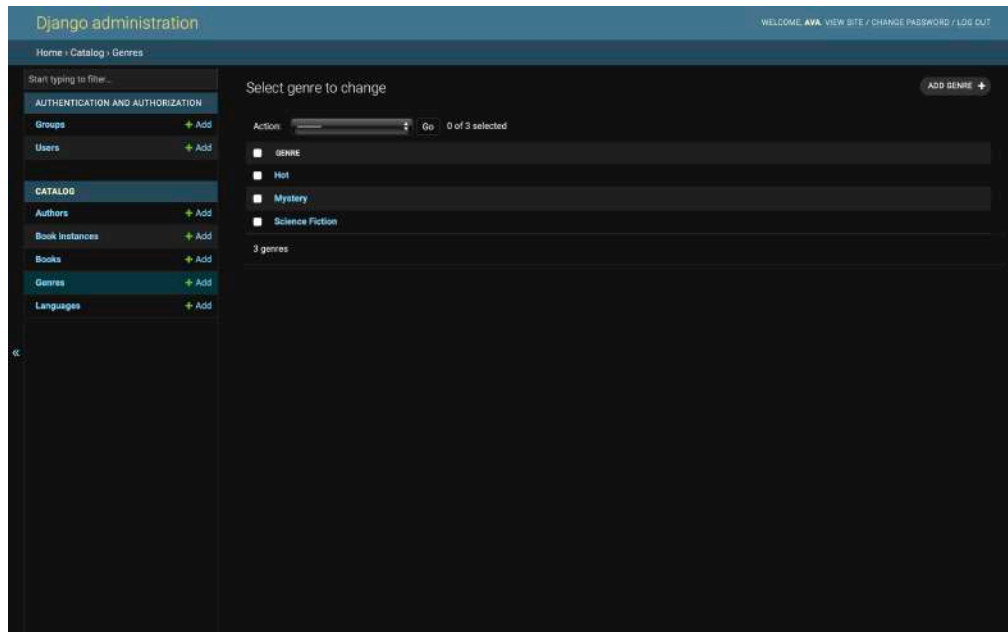
☒

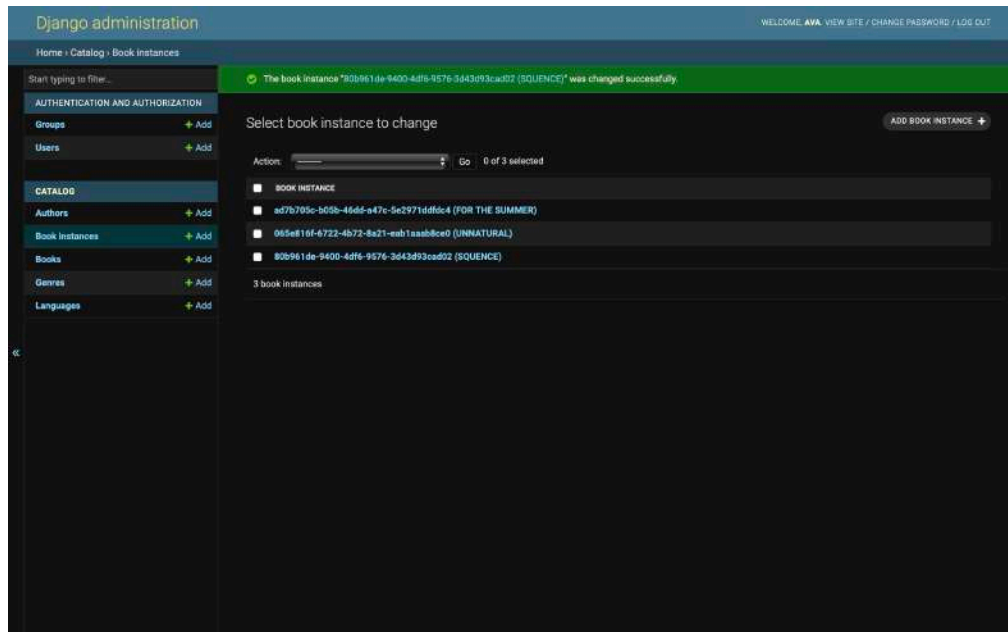
Kim, JiYeon

☒

Lee, LuDa

3 authors





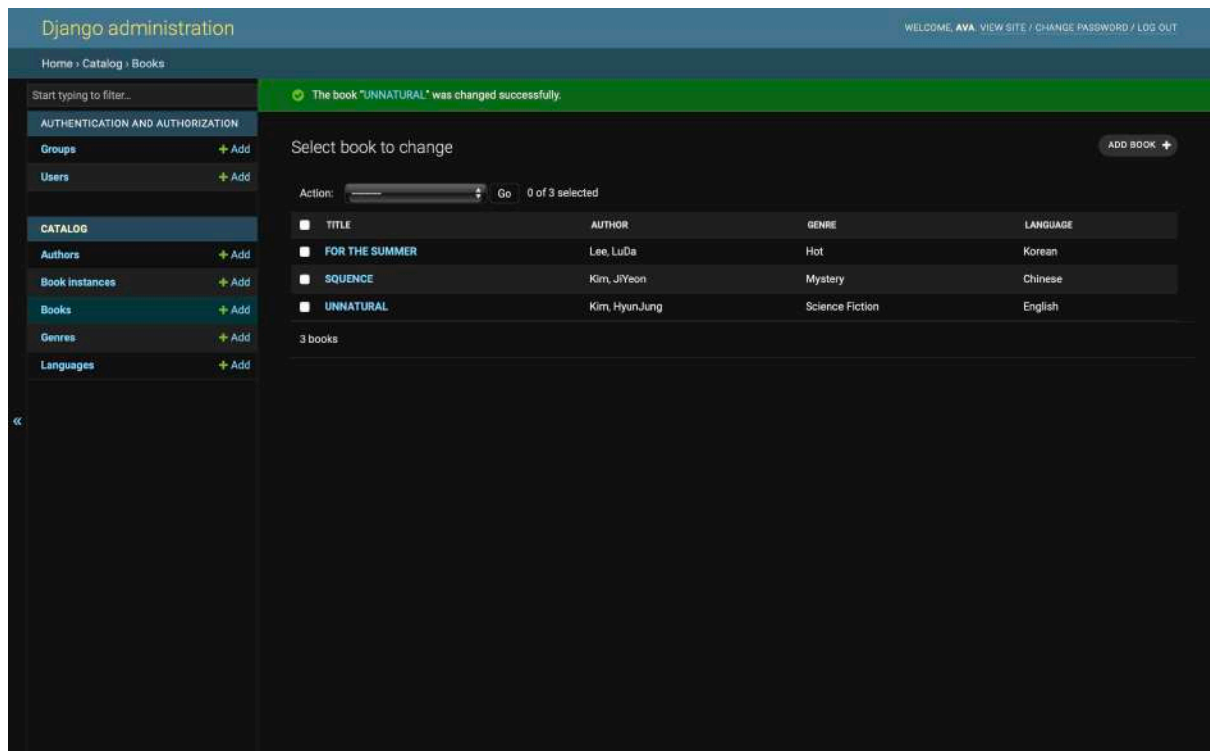
- 註冊一個模型管理類別 (ModelAdmin class)

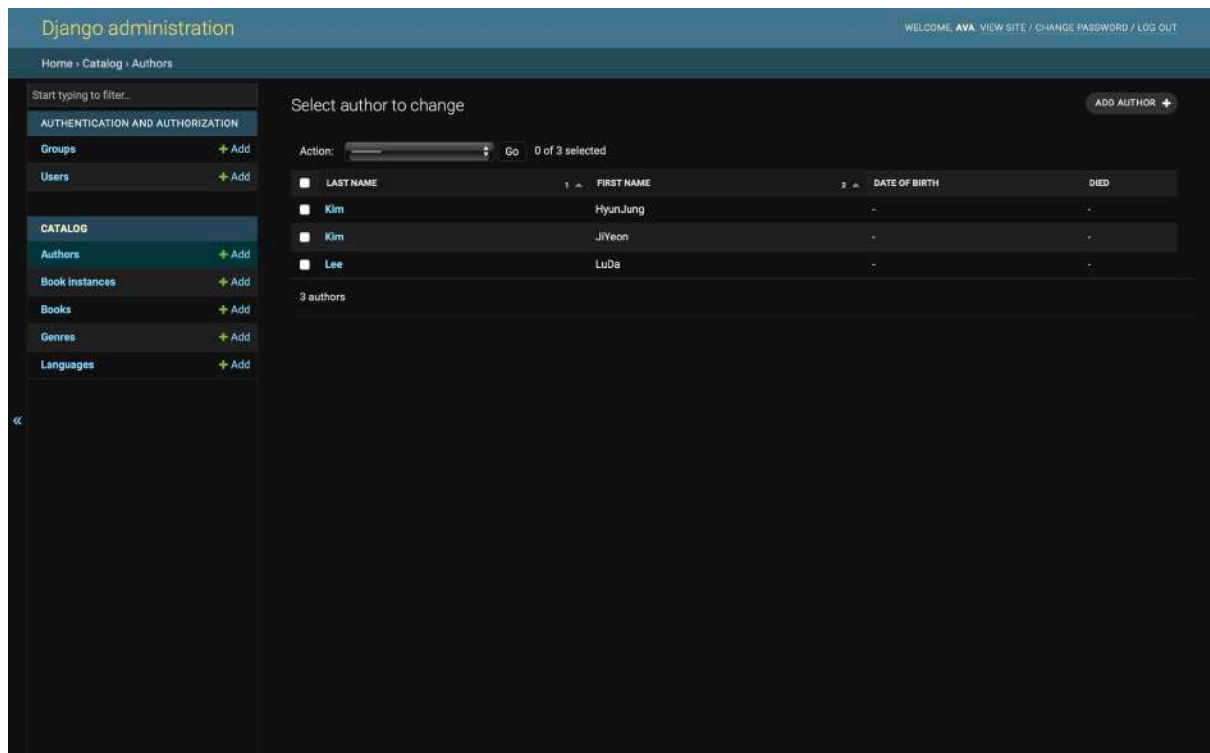
## 改變模型在管理站的陳列方式

```
Users > avachou > django > projects > myworld > locallibrary > locallibrary > catalog > admin.py > BookAdmin
1 from django.contrib import admin
2
3 # Register your models here.
4 from .models import Author, Genre, Book, BookInstance, Language
5
6 #admin.site.register(Book)
7 # admin.site.register(BookInstance)
8
9 #admin.site.register(Author)
10 # Define the admin class
11 class AuthorAdmin(admin.ModelAdmin):
12     list_display = ('last_name', 'first_name', 'date_of_birth', 'date_of_death')
13
14
15
16 admin.site.register(Author, AuthorAdmin)
17 admin.site.register(Genre)
18 admin.site.register(Language)
19
20 # Register the Admin classes for Book using the decorator
21 @admin.register(Book)
22 class BookAdmin(admin.ModelAdmin):
23     list_display = ('title', 'author', 'display_genre', "language")
24
25
26 # Register the Admin classes for BookInstance using the decorator
27 @admin.register(BookInstance)
28 class BookInstanceAdmin(admin.ModelAdmin):
29     pass
30
```



```
Users > avachou > django > projects > myworld > locallibrary > locallibrary > catalog > models.py > Language
22
23 class Book(models.Model):
24     """Model representing a book (but not a specific copy of a book)."""
25     title = models.CharField(max_length=200)
26     author = models.ForeignKey('Author', on_delete=models.SET_NULL, null=True)
27
28     # Foreign Key used because book can only have one author, but authors can have multiple books
29     # Author as a string rather than object because it hasn't been declared yet in the file.
30     summary = models.TextField(max_length=1000, help_text='Enter a brief description of the book')
31     isbn = models.CharField('ISBN', max_length=13, help_text='13 Character <a href="https://www.isbn-international.org/content/what-is-isbn">ISBN')
32
33     # ManyToManyField used because genre can contain many books. Books can cover many genres.
34     # Genre class has already been defined so we can specify the object above.
35     genre = models.ManyToManyField(Genre, help_text='Select a genre for this book')
36     language = models.ForeignKey('Language', on_delete=models.SET_NULL, null=True)
37
38     def __str__(self):
39         """String for representing the Model object."""
40         return self.title
41
42     def get_absolute_url(self):
43         """Returns the url to access a detail record for this book."""
44         return reverse('book-detail', args=[str(self.id)])
45
46     def display_genre(self):
47         """Create a string for the Genre. This is required to display genre in Admin."""
48         return ', '.join(genre.name for genre in self.genre.all()[:3])
49
50     display_genre.short_description = 'Genre'
51
52
53 import uuid # Required for unique book instances
54
55 class BookInstance(models.Model):
56     """Model representing a specific copy of a book (i.e. that can be borrowed from the library)."""
```



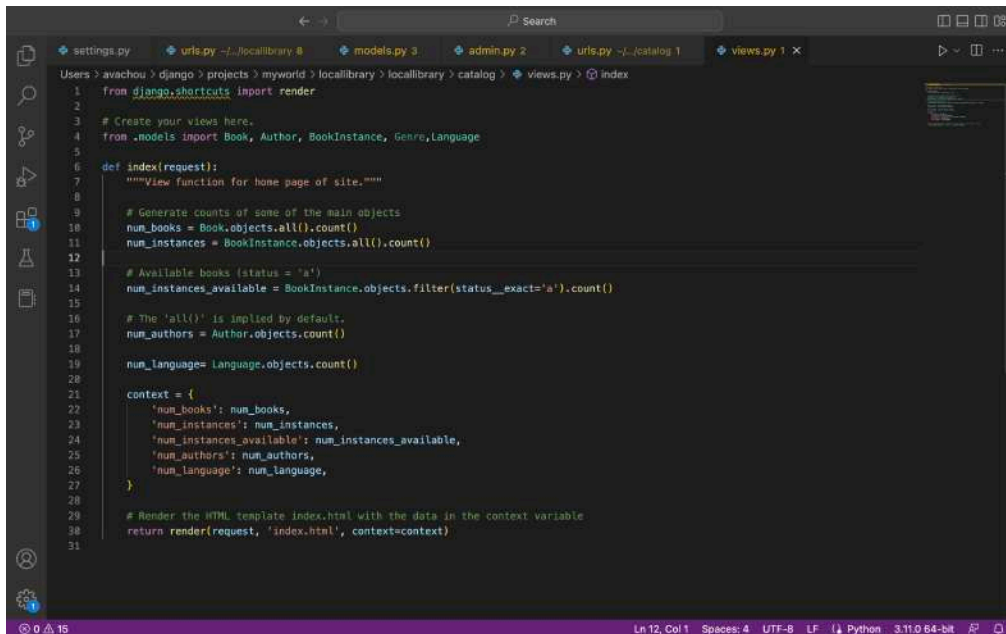


## 創建索引頁面

- View

用來處理 HTTP 請求的函式，根據需求從資料庫取得資料，通過使用 HTML 模板呈現此數據來生成 HTML，並且在一個 HTTP 回應中返回 HTML 來呈現給用戶





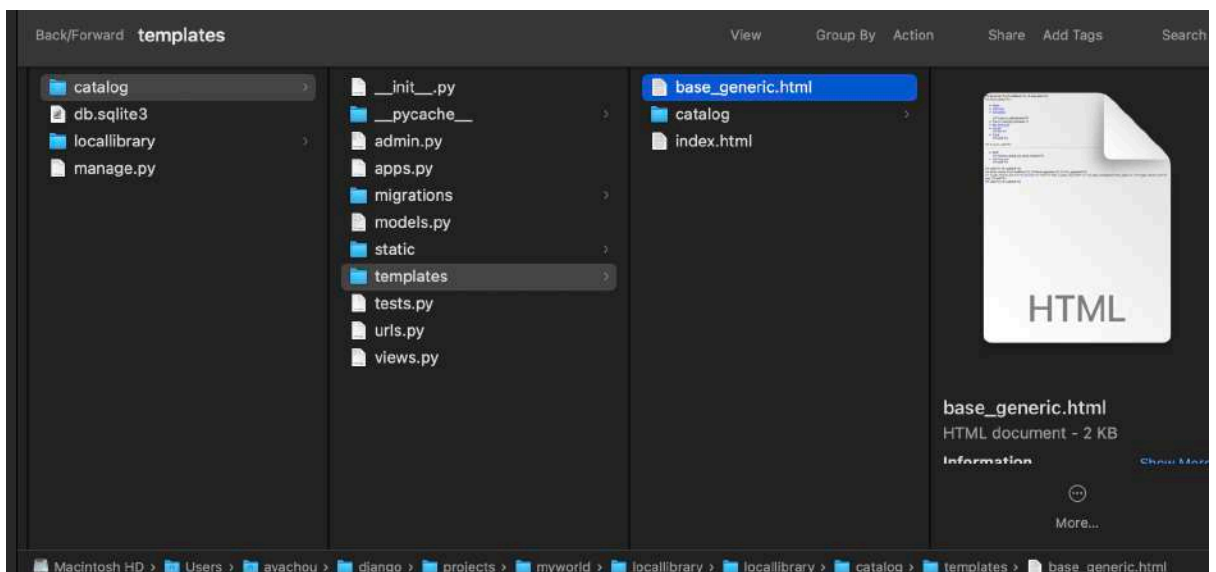
```
1 from django.shortcuts import render
2
3 # Create your views here.
4 from .models import Book, Author, BookInstance, Genre, Language
5
6 def index(request):
7     """View function for home page of site."""
8
9     # Generate counts of some of the main objects
10    num_books = Book.objects.all().count()
11    num_instances = BookInstance.objects.all().count()
12
13    # Available books (status = 'a')
14    num_instances_available = BookInstance.objects.filter(status__exact='a').count()
15
16    # The 'all()' is implied by default.
17    num_authors = Author.objects.count()
18
19    num_language = Language.objects.count()
20
21    context = {
22        'num_books': num_books,
23        'num_instances': num_instances,
24        'num_instances_available': num_instances_available,
25        'num_authors': num_authors,
26        'num_language': num_language,
27    }
28
29    # Render the HTML template index.html with the data in the context variable
30    return render(request, 'index.html', context=context)
```

- Template

一個文本文件，用於定義文件（例如 HTML 頁面）的結構或佈局，並使用佔位符表示實際內容

- 創建LocalLibrary網站的基本模板

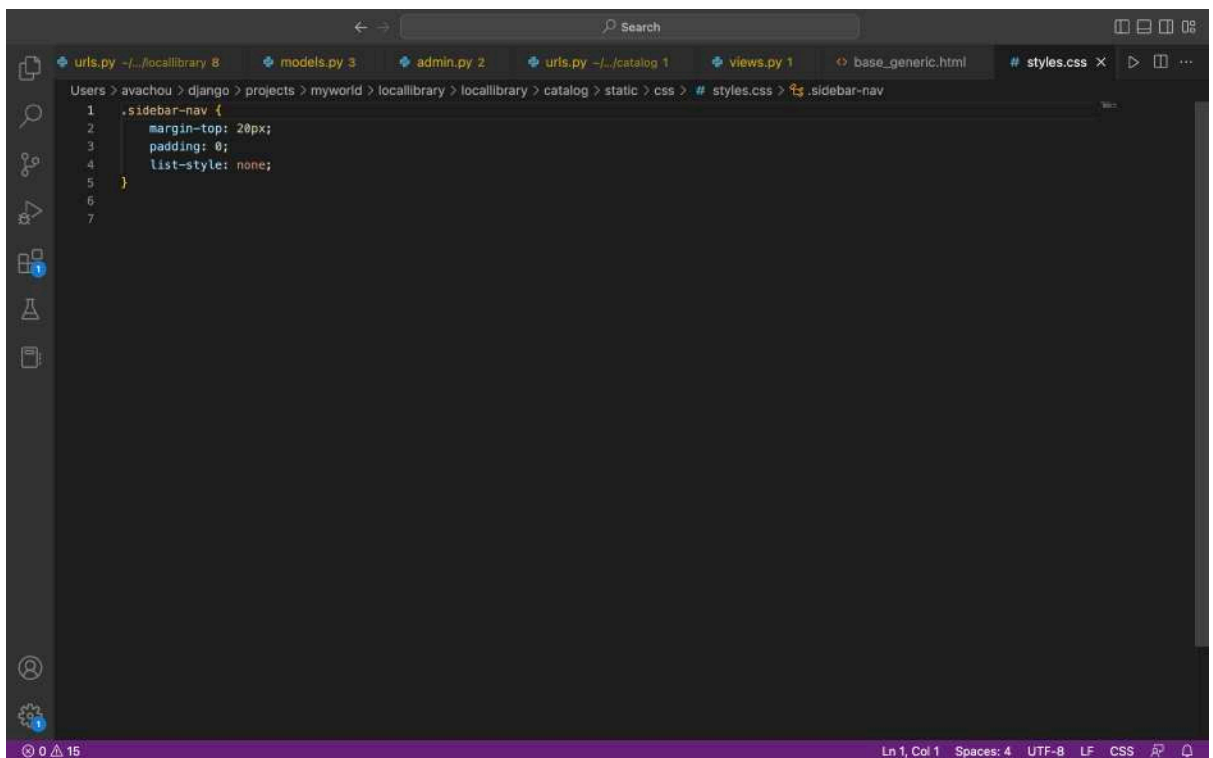
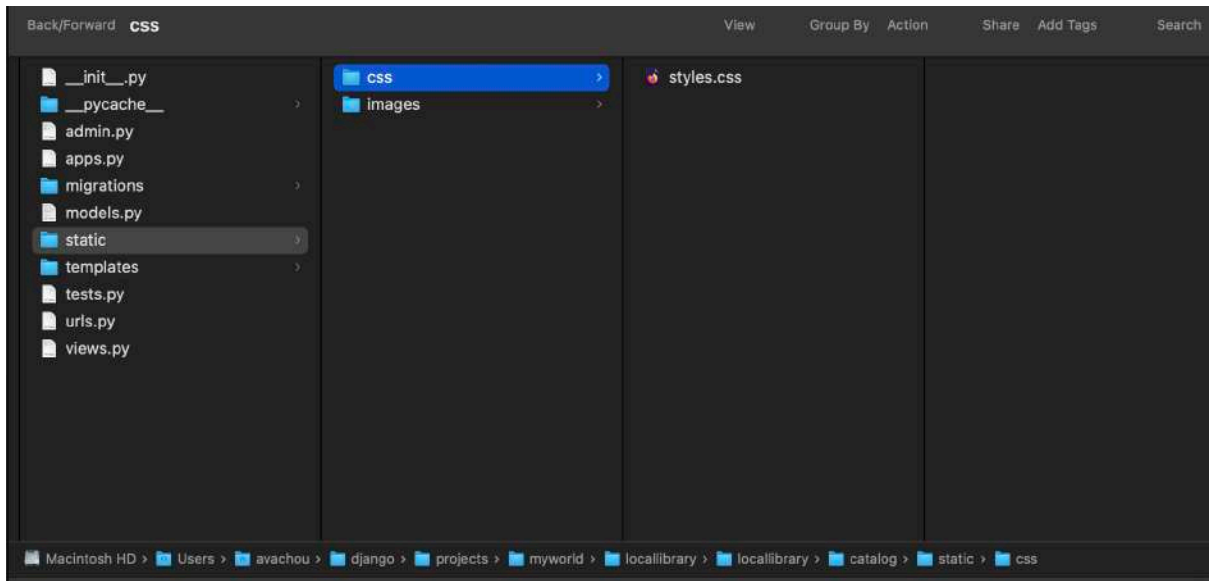
base\_generic.html



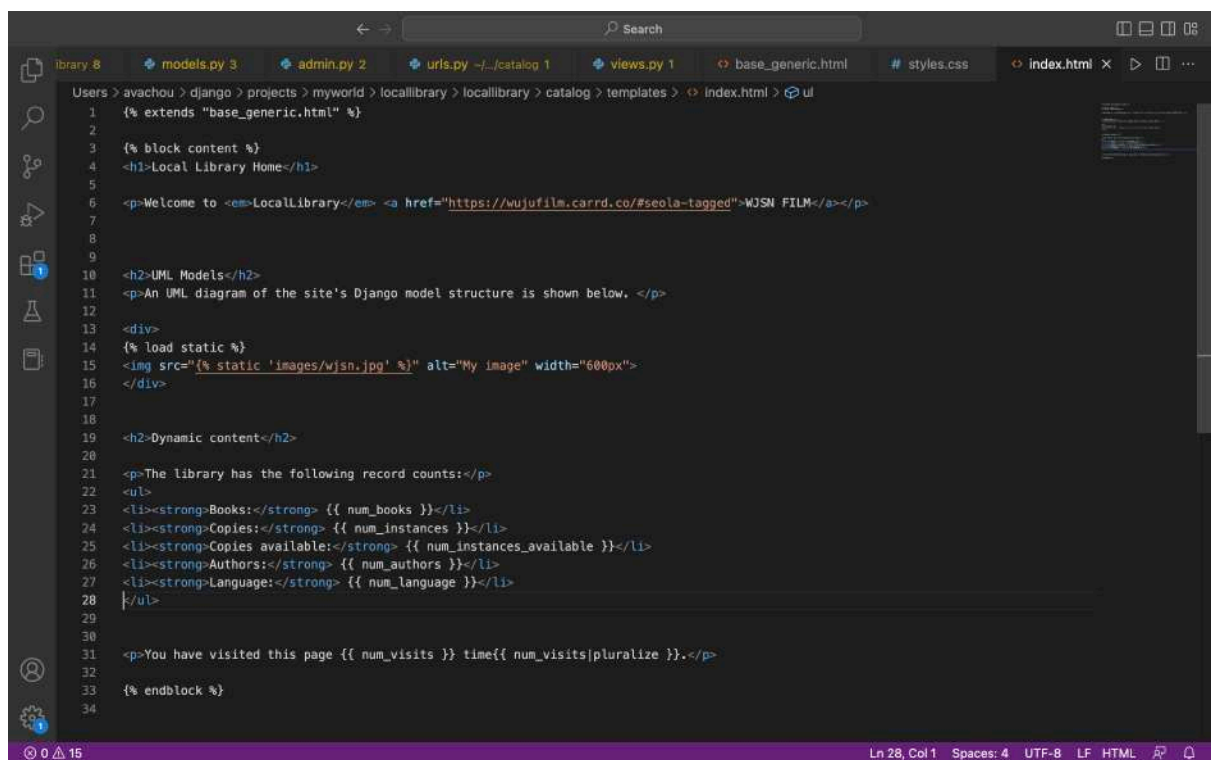
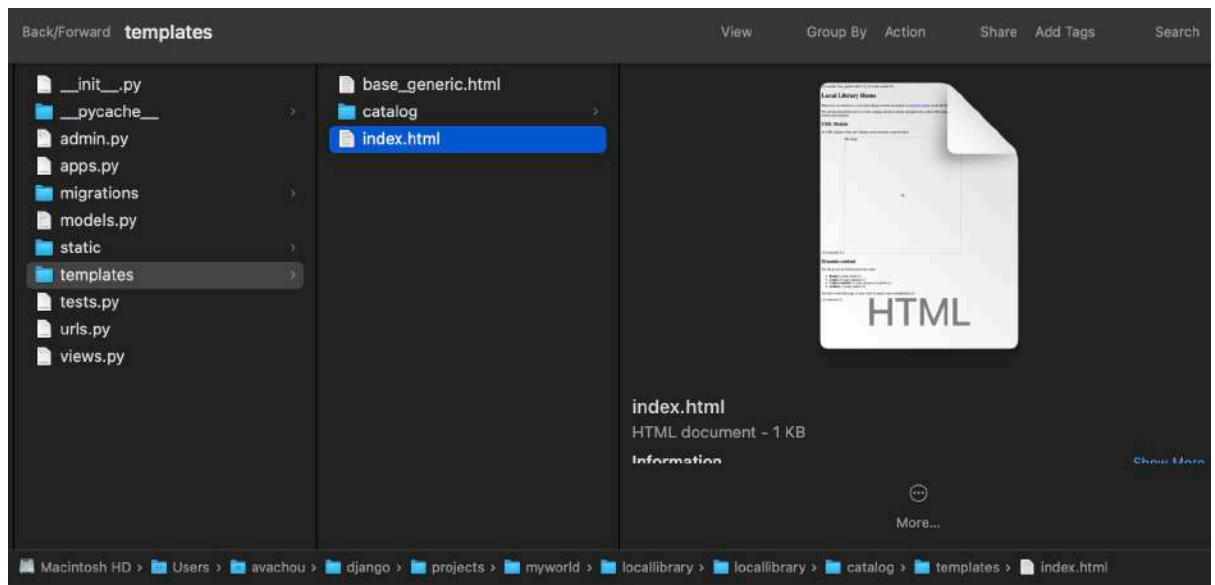
```
36 </ul>
37
38 {% if user.is_staff %}
39 <hr>
40 <ul class="sidebar-nav">
41 <li>Staff</li>
42 {% if perms.catalog.can_mark_returned %}
43 <li><a href="{% url 'all-borrowed' %}">All borrowed</a></li>
44 {% endif %}
45 </ul>
46 {% endif %}
47
48 {% endblock %}
49 </div>
50 <div class="col-sm-10">
51 {% block content %}{% endblock %}
52
53 {% block pagination %}
54 {% if is_paginated %}
55 <div class="pagination">
56 <span class="page-links">
57 {% if page_obj.has_previous %}
58 <a href="{% request.path %}?page={{ page_obj.previous_page_number }}">previous</a>
59 {% endif %}
60 <span class="page-current">
61 Page {{ page_obj.number }} of {{ page_obj.paginator.num_pages }}.
62 </span>
63 {% if page_obj.has_next %}
64 <a href="{% request.path %}?page={{ page_obj.next_page_number }}">next</a>
65 {% endif %}
66 </span>
67 </div>
68 {% endif %}
69 {% endblock %}
70
```

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4
5 {% block title %}<title>Local Library</title>{% endblock %}
6 <meta charset="utf-8">
7 <meta name="viewport" content="width=device-width, initial-scale=1">
8 <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-18mE4kqBq781"
9
10 <!-- Add additional CSS in static file -->
11 {% load static %}
12 <link rel="stylesheet" href="{% static 'css/styles.css' %}">
13 </head>
14 <body>
15
16 <div class="container-fluid">
17
18 <div class="row">
19 <div class="col-sm-2">
20 {% block sidebar %}
21 <ul class="sidebar-nav">
22 <li><a href="{% url 'index' %}">Home</a></li>
23 <li><a href="{% url 'books' %}">All books</a></li>
24 <li><a href="{% url 'authors' %}">All authors</a></li>
25 </ul>
26
27 <ul class="sidebar-nav">
28 {% if user.is_authenticated %}
29 <li>User: {{ user.get_username }}</li>
30 <li><a href="{% url 'my-borrowed' %}">My Borrowed</a></li>
31 <li><a href="{% url 'logout' %}?next={{ request.path }}">Logout</a></li>
32 {% else %}
33 <li><a href="{% url 'login' %}?next={{ request.path }}">Login</a></li>
34 {% endif %}
35
```

styles.css



index.html



[Home](#)  
[All books](#)  
[All authors](#)

## Local Library Home

Welcome to *LocalLibrary*  
[WJSN FILM](#)



### Dynamic content

The library has the following record counts:

- **Books:** 3
- **Copies:** 3
- **Copies available:** 1
- **Authors:** 3
- **Language:** 3

- Book list page

## 顯示書籍庫列表

[Home](#)  
[All books](#)  
[All authors](#)

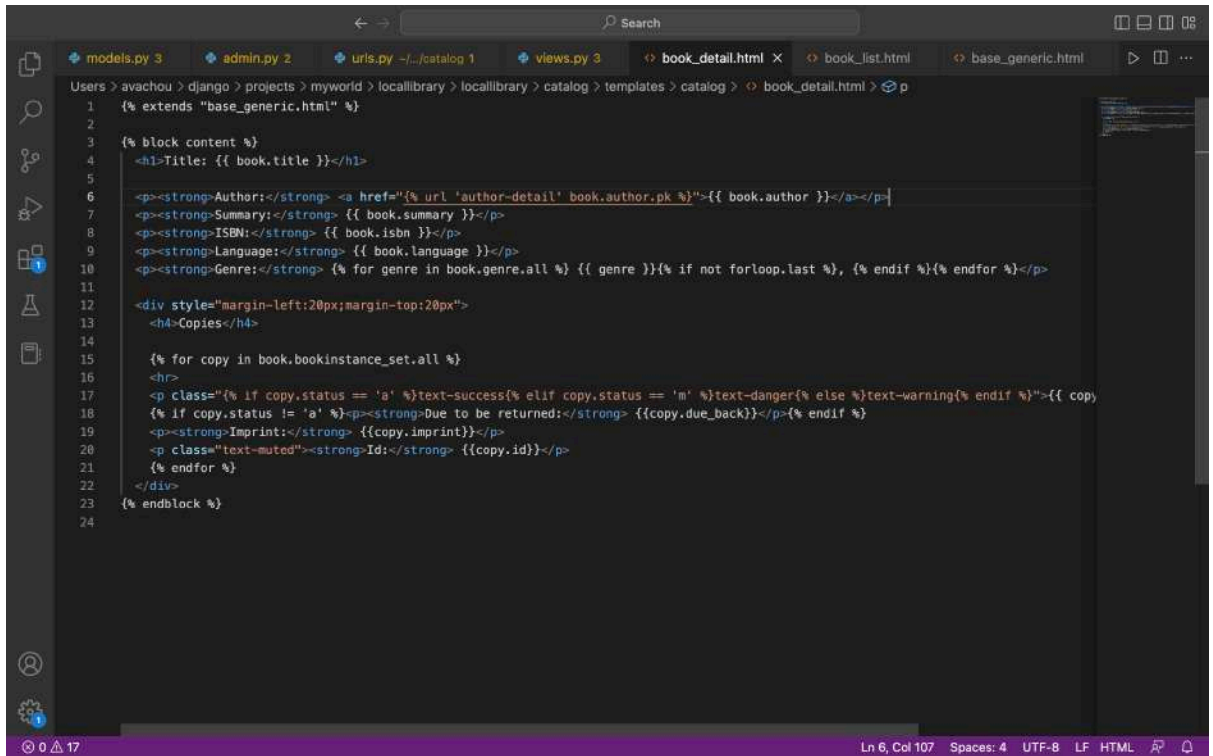
## Book List

- [UNNATURAL](#) (Kim, HyunJung)
- [SEQUENCE](#) (Kim, JiYeon)
- [FOR THE SUMMER](#) (Lee, LuDa)
- [NEVERLAND](#) (Kim, HyunJung)

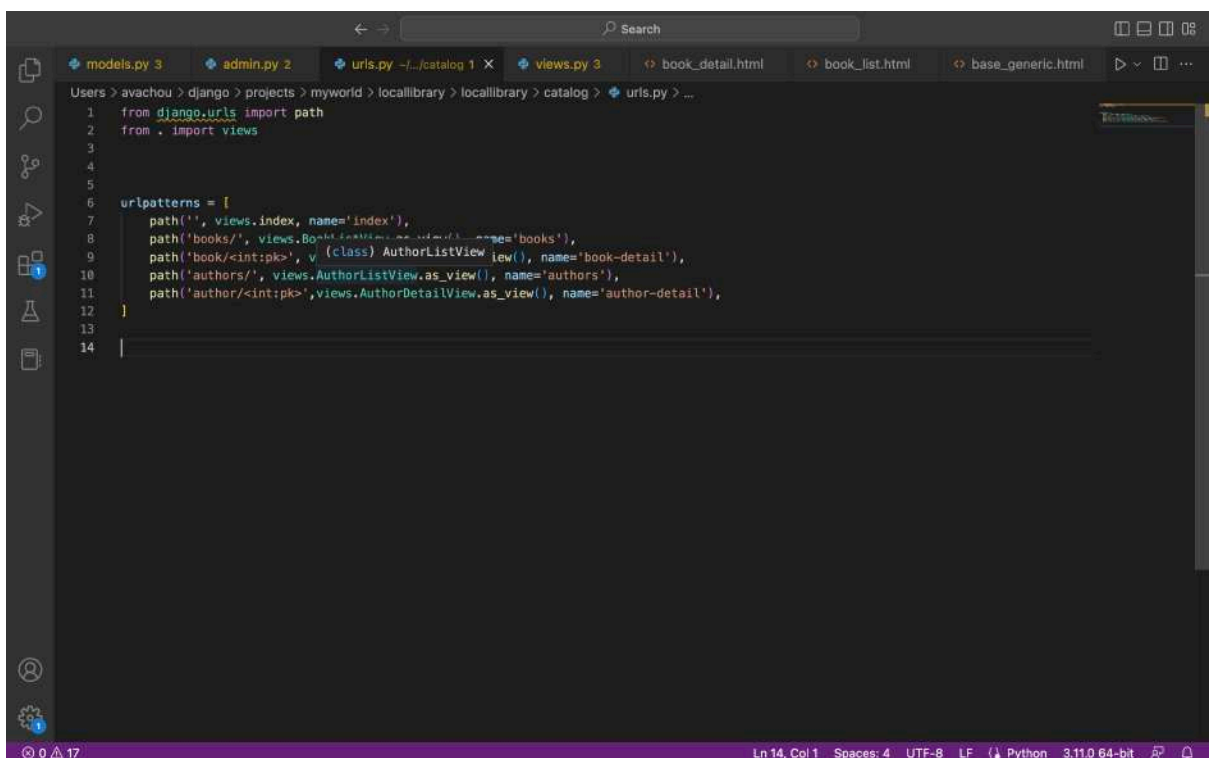


- Book detail page

顯示各別書籍的資訊



```
1 {% extends "base_generic.html" %}
2
3 {% block content %}
4 <h1>Title: {{ book.title }}</h1>
5
6 <p><strong>Author:</strong> <a href="{% url 'author-detail' book.author.pk %}">{{ book.author }}</a></p>
7 <p><strong>Summary:</strong> {{ book.summary }}</p>
8 <p><strong>ISBN:</strong> {{ book.isbn }}</p>
9 <p><strong>Language:</strong> {{ book.language }}</p>
10 <p><strong>Genre:</strong> {% for genre in book.genre.all %} {{ genre }}{% if not forloop.last %}, {% endif %}{% endfor %}</p>
11
12 <div style="margin-left:20px;margin-top:20px">
13 <h4>Copies</h4>
14
15 {% for copy in book.bookinstance_set.all %}
16 <hr>
17 <p class="{% if copy.status == 'a' %}text-success{% elif copy.status == 'm' %}text-danger{% else %}text-warning{% endif %}">{{ copy
18 {% if copy.status != 'a' %}<strong>Due to be returned:</strong> {{copy.due_back}}</p>{% endif %}
19 <p><strong>Imprint:</strong> {{copy.imprint}}</p>
20 <p class="text-muted"><strong>Id:</strong> {{copy.id}}</p>
21 {% endfor %}
22 </div>
23 {% endblock %}
24
```



```
1 from django.urls import path
2 from . import views
3
4
5
6 urlpatterns = [
7     path('', views.index, name='index'),
8     path('books/', views.BookListView.as_view(), name='books'),
9     path('book/<int:pk>', views.AuthorDetailView.as_view(), name='book-detail'),
10    path('authors/', views.AuthorListView.as_view(), name='authors'),
11    path('author/<int:pk>', views.AuthorDetailView.as_view(), name='author-detail'),
12 ]
13
14
```

[Home](#)  
[All books](#)  
[All authors](#)

## Title: UNNATURAL

**Author:** [Kim, HyunJung](#)

**Summary:** WJSN - UNNATURAL (9TH MINI ALBUM)

**ISBN:** 0000020210408

**Language:** English

**Genre:** Sci-Fi

### Copies

**Maintenance**

**Due to be returned:** None

**Imprint:** STARSHIP ENTERTAINMENT

**Id:** 065e816f-6722-4b72-8a21-eab1aaab8ce0

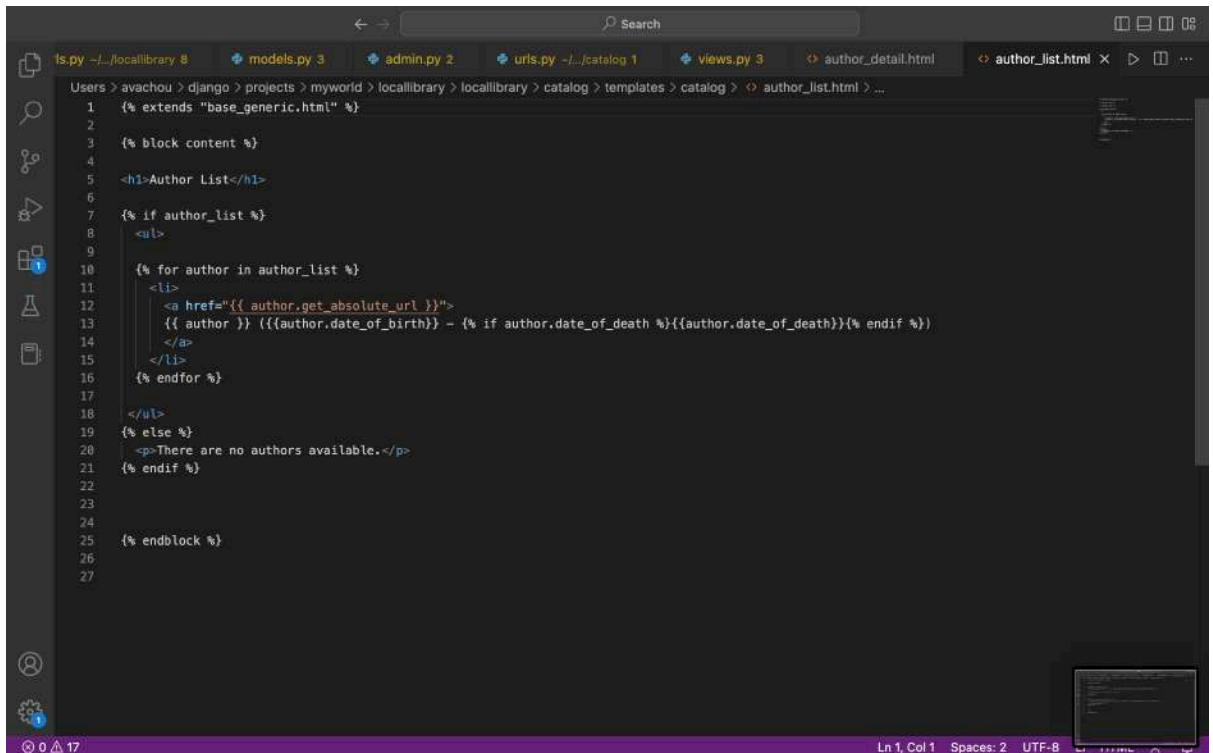
- Author list

## 顯示作者列表

[Home](#)  
[All books](#)  
[All authors](#)

## Author List

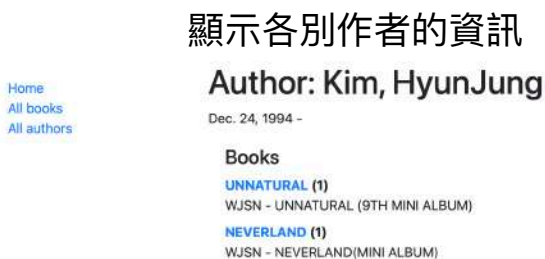
- [Kim, HyunJung](#) (Dec. 24, 1994 - )
- [Kim, JiYeon](#) (Aug. 19, 1995 - )
- [Lee, LuDa](#) (Feb. 6, 1997 - )



The screenshot shows a code editor with the file 'author\_list.html' open. The code is a Django template that extends 'base\_generic.html'. It contains a block for the author list, which is rendered as an unordered list. Each list item contains a link to the author's detail page, followed by the author's name and birth/death dates. If no authors are available, a message is displayed.

```
1 {% extends "base_generic.html" %}
2
3 {% block content %}
4
5 <h1>Author List</h1>
6
7 {% if author_list %}
8   <ul>
9
10     {% for author in author_list %}
11       <li>
12         <a href="{% author.get_absolute_url %}">
13           {{ author }} ({{author.date_of_birth}} - {% if author.date_of_death %}{{author.date_of_death}}{% endif %})
14         </a>
15       </li>
16     {% endfor %}
17   </ul>
18 {% else %}
19   <p>There are no authors available.</p>
20 {% endif %}
21
22 {% endblock %}
23
24
25
26
27
```

- Author detail



The screenshot shows a web application interface for an author's details. The page title is 'Author: Kim, HyunJung'. Below the title, the birth date 'Dec. 24, 1994 -' is displayed. A section titled 'Books' lists two albums: 'UNNATURAL (1)' and 'NEVERLAND (1)', both by WJSN. The left sidebar contains links for 'Home', 'All books', and 'All authors'.

Home  
All books  
All authors

## 顯示各別作者的資訊

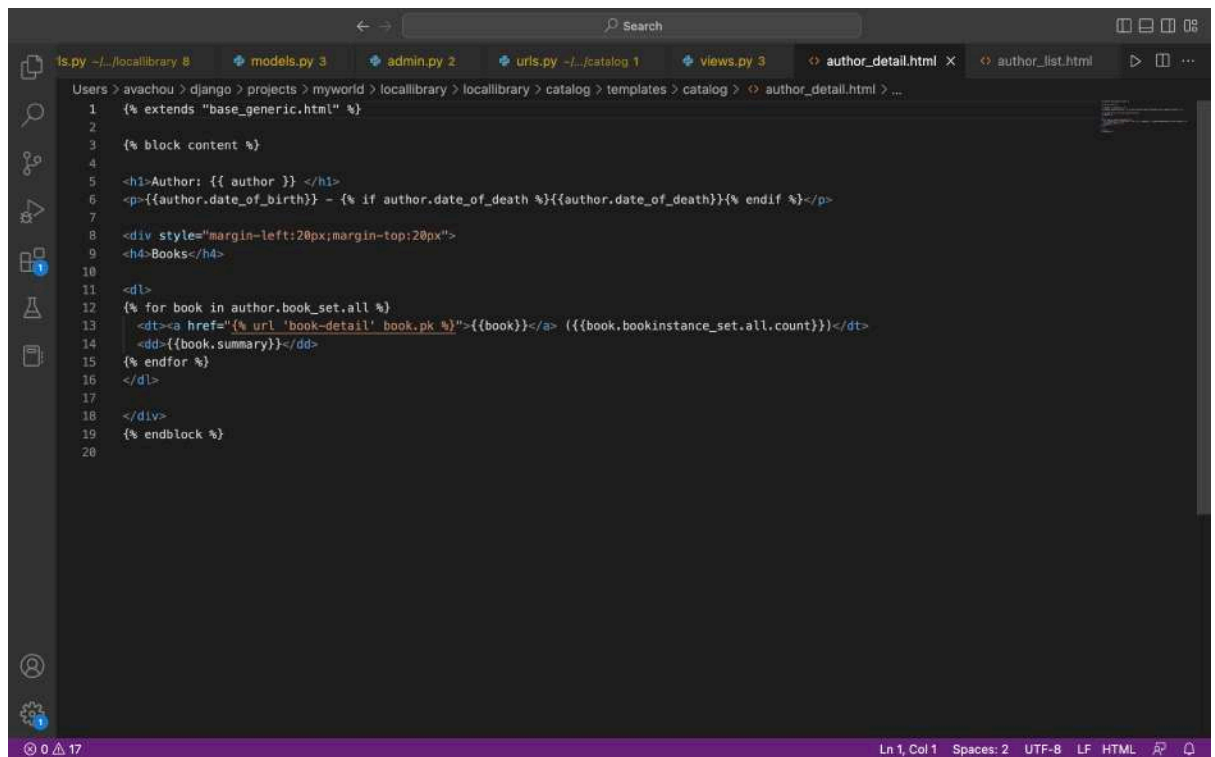
### Author: Kim, HyunJung

Dec. 24, 1994 -

#### Books

- [UNNATURAL \(1\)](#)  
WJSN - UNNATURAL (9TH MINI ALBUM)
- [NEVERLAND \(1\)](#)  
WJSN - NEVERLAND(MINI ALBUM)

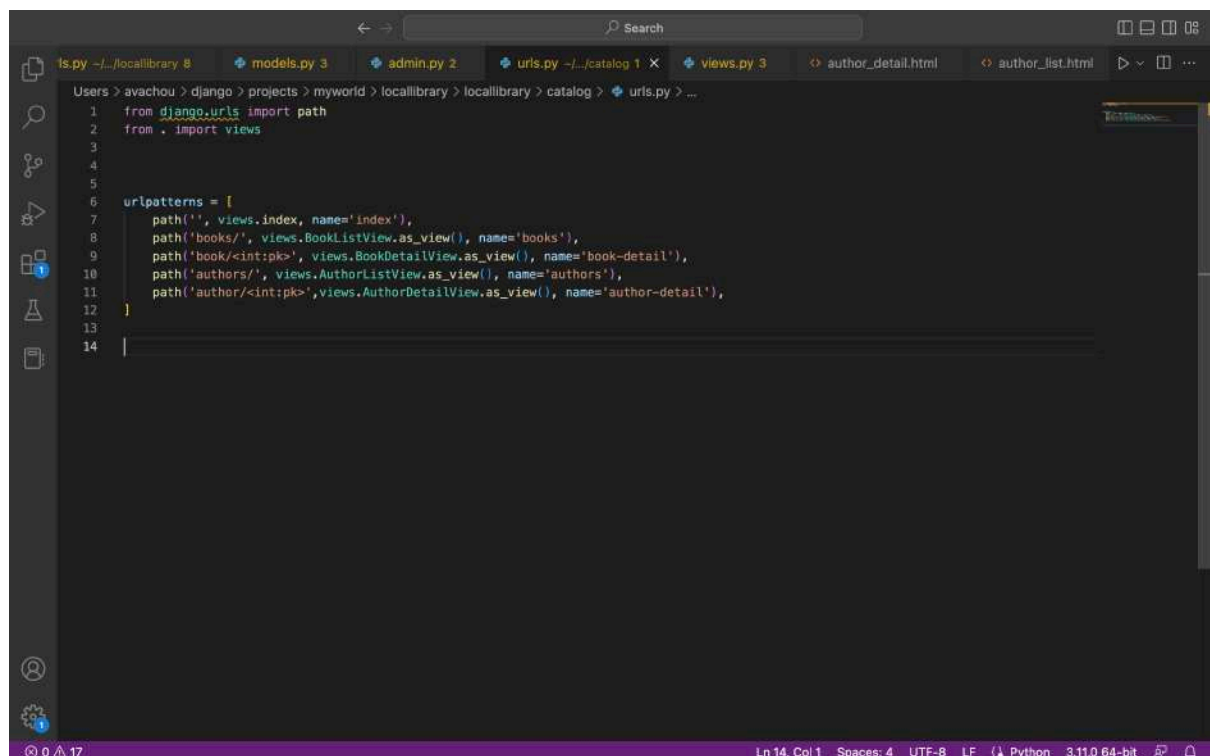




A screenshot of a code editor window showing the `author_detail.html` template. The editor has a dark theme and a sidebar on the left with icons for file explorer, search, and other tools. The top bar shows the file path: `Users > avachou > django > projects > myworld > locallibrary > locallibrary > catalog > templates > catalog > author_detail.html > ...`. The code is as follows:

```
1 {% extends "base_generic.html" %}
2
3 {% block content %}
4
5 <h1>Author: {{ author }} </h1>
6 <p>{{ author.date_of_birth }} - {% if author.date_of_death %}{{ author.date_of_death }}{% endif %}</p>
7
8 <div style="margin-left:20px;margin-top:20px">
9   <h4>Books</h4>
10
11   <div>
12     {% for book in author.book_set.all %}
13       <div><a href="{% url 'book-detail' book.pk %}">{{ book }}</a> ({{ book.bookinstance_set.all.count }})</div>
14       <div>{{ book.summary }}</div>
15     {% endfor %}
16   </div>
17
18 </div>
19 {% endblock %}
20
```

The status bar at the bottom indicates: `Ln 1, Col 1 Spaces: 2 UTF-8 LF HTML`.

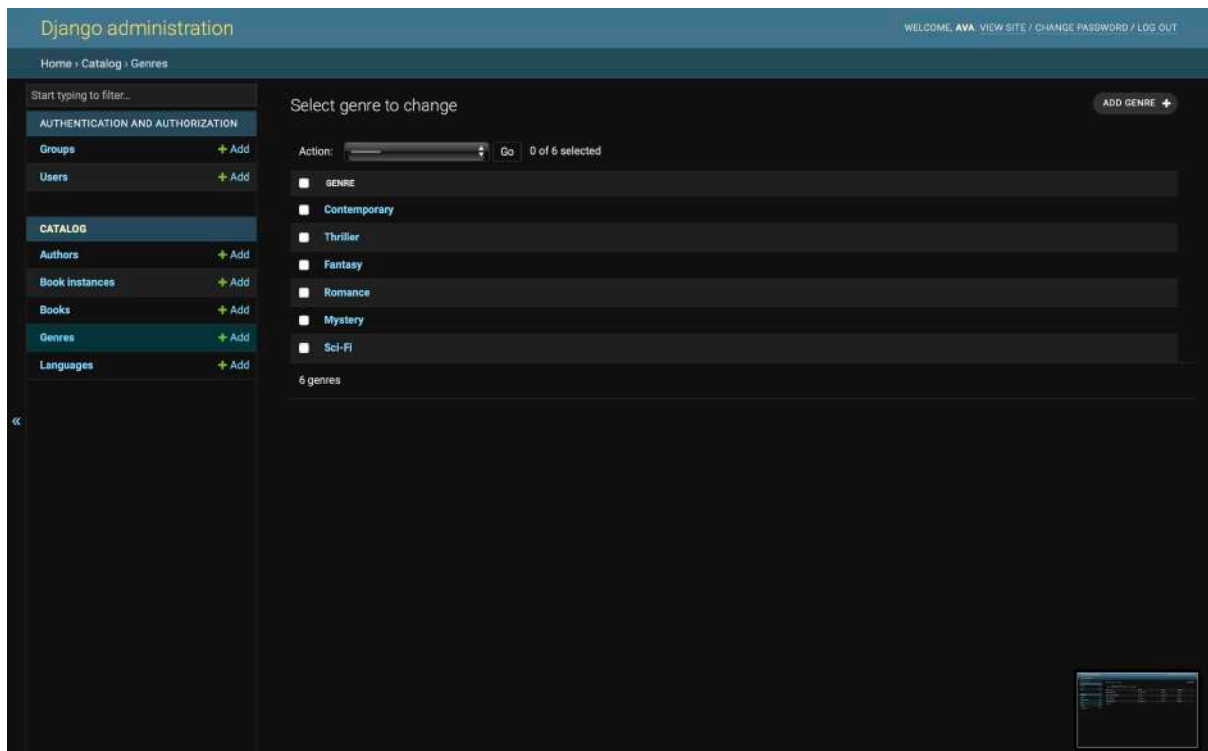
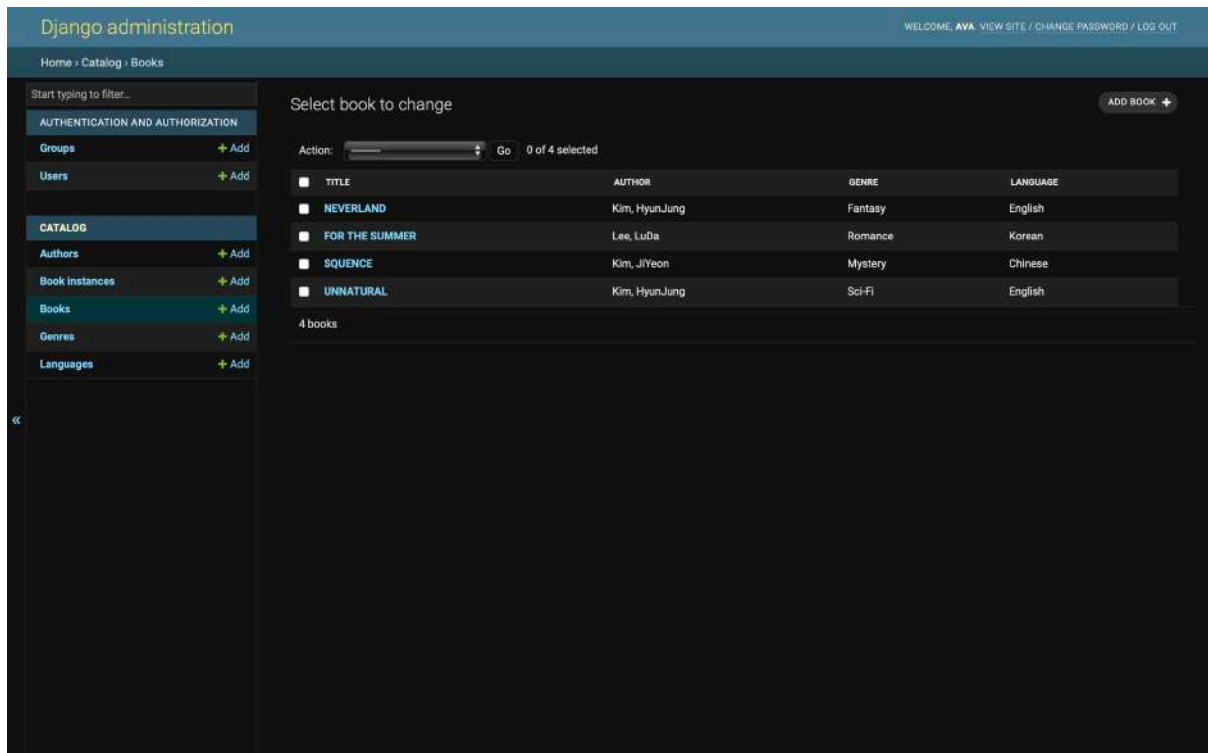


A screenshot of a code editor window showing the `urls.py` file. The editor has a dark theme and a sidebar on the left with icons for file explorer, search, and other tools. The top bar shows the file path: `Users > avachou > django > projects > myworld > locallibrary > locallibrary > catalog > urls.py > ...`. The code is as follows:

```
1 from django.urls import path
2 from . import views
3
4
5
6 urlpatterns = [
7     path('', views.index, name='index'),
8     path('books/', views.BookListView.as_view(), name='books'),
9     path('book/<int:pk>', views.BookDetailView.as_view(), name='book-detail'),
10    path('authors/', views.AuthorListView.as_view(), name='authors'),
11    path('author/<int:pk>', views.AuthorDetailView.as_view(), name='author-detail'),
12 ]
13
14
```

The status bar at the bottom indicates: `Ln 14, Col 1 Spaces: 4 UTF-8 LF Python 3.11.0 64-bit`.

- 在加入Bootstrap將前端美化前之成果



Django administration WELCOME, AVA. [VIEW SITE](#) / [CHANGE PASSWORD](#) / [LOG OUT](#)

Home > Catalog > Book instances > ad7b705c-b05b-46dd-a47c-5e2971ddfdc4 (FOR THE SUMMER)

Start typing to filter...

**AUTHENTICATION AND AUTHORIZATION**

- Groups [+ Add](#)
- Users [+ Add](#)

**CATALOG**

- Authors [+ Add](#)
- Book instances [+ Add](#)
- Books [+ Add](#)
- Genres [+ Add](#)
- Languages [+ Add](#)

Change book instance

ad7b705c-b05b-46dd-a47c-5e2971ddfdc4 (FOR THE SUMMER) [HISTORY](#)

Book:  [+](#) [-](#) [x](#) [e](#)

Imprint:


Id:   
Unique ID for this particular book across whole library

**Availability**

Status:  [+](#) [-](#)  
Book availability

Due back:  Today | [📅](#)

[Delete](#) [Save and add another](#) [Save and continue editing](#) [SAVE](#)



Django administration WELCOME, AVA. [VIEW SITE](#) / [CHANGE PASSWORD](#) / [LOG OUT](#)

Home > Catalog > Authors > Kim, JiYeon

Start typing to filter...

**AUTHENTICATION AND AUTHORIZATION**

- Groups [+ Add](#)
- Users [+ Add](#)

**CATALOG**

- Authors [+ Add](#)
- Book instances [+ Add](#)
- Books [+ Add](#)
- Genres [+ Add](#)
- Languages [+ Add](#)

Change author


Kim, JiYeon [HISTORY](#) [VIEW ON SITE](#)

First name:

Last name:

Date of birth:  Today | [📅](#) Died:  Today | [📅](#)

[Delete](#) [Save and add another](#) [Save and continue editing](#) [SAVE](#)



[Home](#)  
[All books](#)  
[All authors](#)

## Author: Kim, HyunJung



Kim, HyunJung (Dec. 24, 1994 - Dec. 24, 2094)

### Books

[UNNATURAL \(1\)](#)

WJSN - UNNATURAL (9TH MINI ALBUM)

[NEVERLAND \(1\)](#)

WJSN - NEVERLAND (MINI ALBUM)

[Home](#)  
[All books](#)  
[All authors](#)

## Title: UNNATURAL

**Author:** [Kim, HyunJung](#)

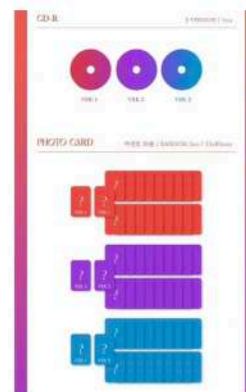
**Summary:** WJSN - UNNATURAL (9TH MINI ALBUM)

**ISBN:** 0000020210408

**Language:** English

**Genre:** Sci-Fi

### Preview



### Copies

[Home](#)  
[All books](#)  
[All authors](#)

## Local Library Home



The library has the following record counts:

- Books: 4
- Copies: 4
- Copies available: 2
- Authors: 3
- Language: 5

[Home](#)  
[All books](#)  
[All authors](#)

## Author List

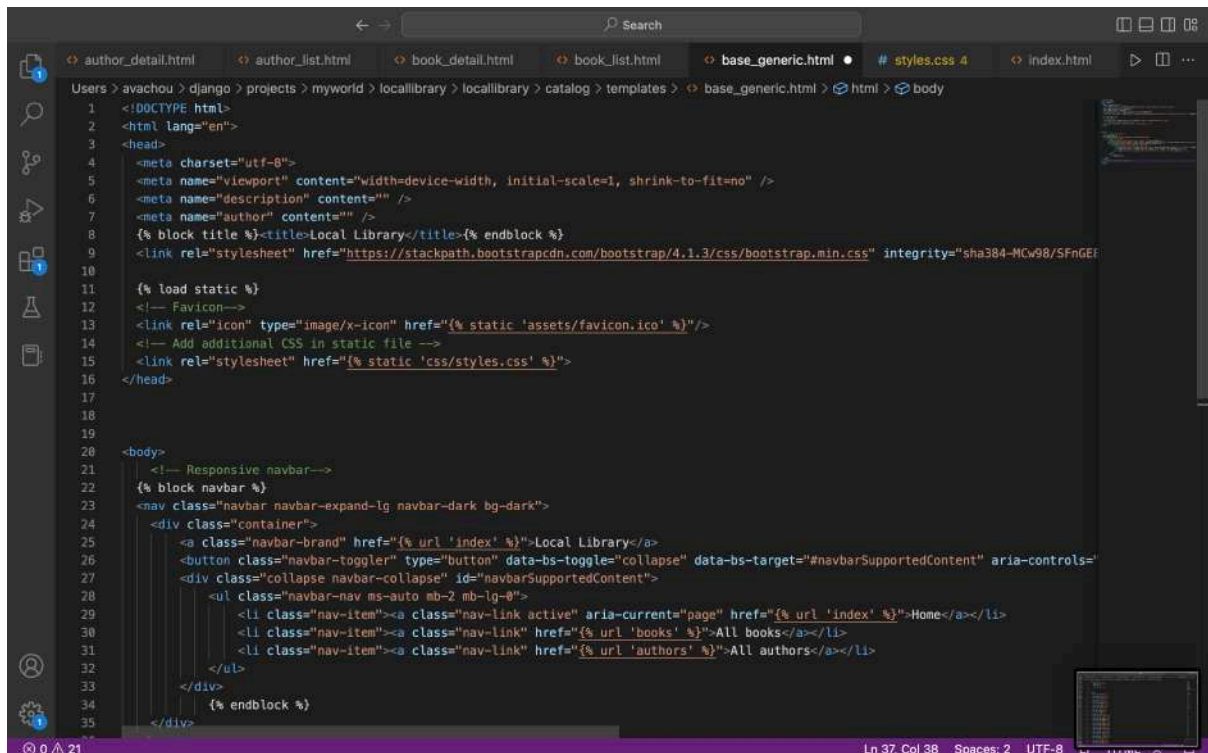
- [Kim, HyunJung](#) (Dec. 24, 1994 - Dec. 24, 2094)
- [Kim, JiYeon](#) (Aug. 19, 1995 - Aug. 19, 2095)
- [Lee, LuDa](#) (Feb. 6, 1997 - Feb. 6, 2097)

[Home](#)  
[All books](#)  
[All authors](#)

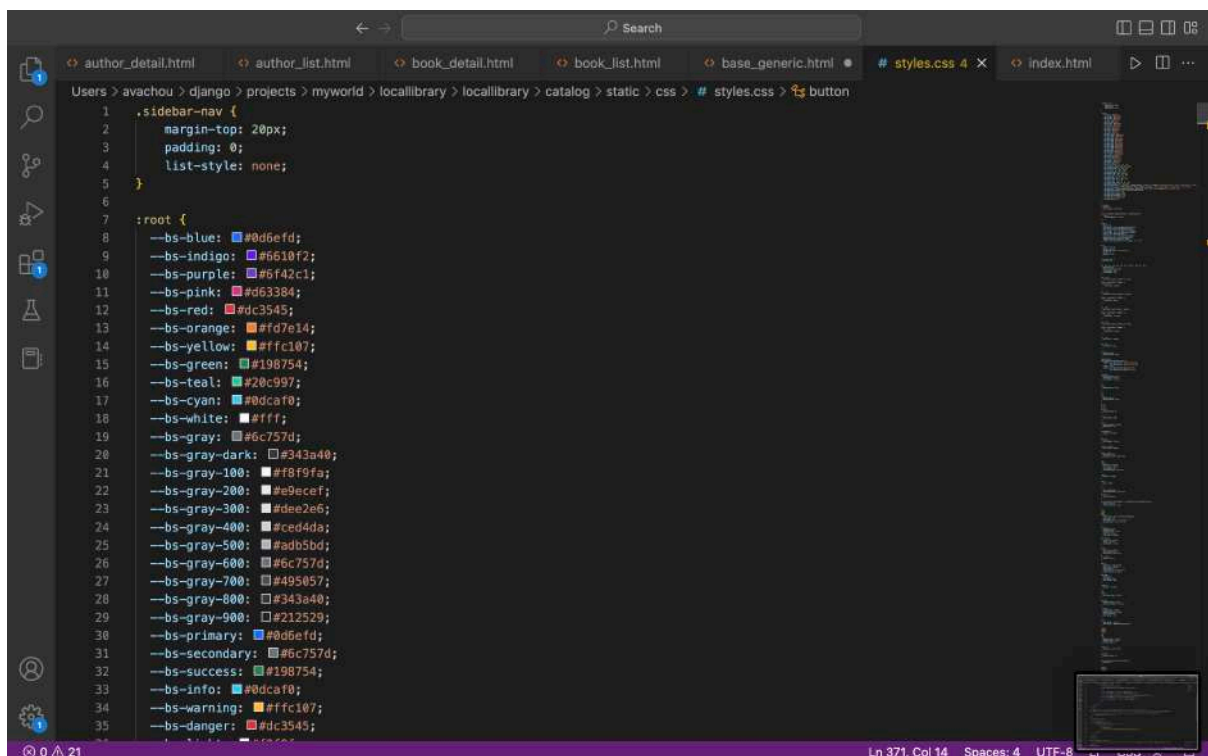
## Book List

- [UNNATURAL](#) (Kim, HyunJung)
- [SEQUENCE](#) (Kim, JiYeon)
- [FOR THE SUMMER](#) (Lee, LuDa)
- [NEVERLAND](#) (Kim, HyunJung)

- 加入Bootstrap



```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="utf-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no" />
6   <meta name="description" content="" />
7   <meta name="author" content="" />
8   {% block title %}<title>Local Library</title>{% endblock %}
9   <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css" integrity="sha384-MCw98/SFnGEI"
10
11   {% load static %}
12   <!-- Favicon -->
13   <link rel="icon" type="image/x-icon" href="{% static 'assets/favicon.ico' %}" />
14   <!-- Add additional CSS in static file -->
15   <link rel="stylesheet" href="{% static 'css/styles.css' %}" />
16 </head>
17
18 <body>
19
20 <!-- Responsive navbar -->
21 {% block navbar %}
22 <nav class="navbar navbar-expand-lg navbar-dark bg-dark">
23   <div class="container">
24     <a class="navbar-brand" href="{% url 'index' %}">Local Library</a>
25     <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarSupportedContent" aria-controls="
26     <div class="collapse navbar-collapse" id="navbarSupportedContent">
27       <ul class="navbar-nav ms-auto mb-2 mb-lg-0">
28         <li class="nav-item"><a class="nav-link active" aria-current="page" href="{% url 'index' %}">Home</a></li>
29         <li class="nav-item"><a class="nav-link" href="{% url 'books' %}">All books</a></li>
30         <li class="nav-item"><a class="nav-link" href="{% url 'authors' %}">All authors</a></li>
31       </ul>
32     </div>
33   </div>
34   {% endblock %}
35 </div>
```



```
1 .sidebar-nav {
2   margin-top: 20px;
3   padding: 0;
4   list-style: none;
5 }
6
7 :root {
8   --bs-blue: #0d6efd;
9   --bs-indigo: #6610f2;
10  --bs-purple: #6f42c1;
11  --bs-pink: #d63384;
12  --bs-red: #dc3545;
13  --bs-orange: #fd7e14;
14  --bs-yellow: #ffc107;
15  --bs-green: #198754;
16  --bs-teal: #20c997;
17  --bs-cyan: #0dcaf0;
18  --bs-white: #fff;
19  --bs-gray: #6c757d;
20  --bs-gray-dark: #343a40;
21  --bs-gray-100: #f8f9fa;
22  --bs-gray-200: #e9ecef;
23  --bs-gray-300: #dee2e6;
24  --bs-gray-400: #ced4da;
25  --bs-gray-500: #adb5bd;
26  --bs-gray-600: #6c757d;
27  --bs-gray-700: #495057;
28  --bs-gray-800: #343a40;
29  --bs-gray-900: #212529;
30  --bs-primary: #0d6efd;
31  --bs-secondary: #6c757d;
32  --bs-success: #198754;
33  --bs-info: #0dcaf0;
34  --bs-warning: #ffc107;
35  --bs-danger: #dc3545;
```



```
← → Search
author_detail.html author_list.html book_detail.html book_list.html base_generic.html # styles.css 4 index.html x ...
acts > myworld > locallibrary > locallibrary > catalog > templates > index.html > section.py-5 > div.container.my-5 > div.row.justify-content-center > div.col-lg-6 > p.lead
1 {% extends "base_generic.html" %}
2
3 {% block content %}
4
5 <header class="py-5 bg-image-full" style="background-image: url('https://source.unsplash.com/wfh8dDlNF0k/1600x900')">
6   <div class="text-center my-5">
7     {% load static %}
8     
9     <h1 class="text-white fs-3 fw-bolder">Local Library Home</h1>
10   </div>
11 </header>
12
13 <!-- Content section -->
14 <section class="py-5">
15   <div class="container my-5">
16     <div class="row justify-content-center">
17       <div class="col-lg-6">
18         <h2>Library Record Counts</h2>
19         <p>The Library has the following record counts:</p>
20         <ul>
21           <li><strong>Books:</strong> {{ num_books }}</li>
22           <li><strong>Copies:</strong> {{ num_instances }}</li>
23           <li><strong>Copies available:</strong> {{ num_instances_available }}</li>
24           <li><strong>Authors:</strong> {{ num_authors }}</li>
25           <li><strong>Language:</strong> {{ num_language }}</li>
26         </ul>
27       </div>
28     </div>
29   </div>
30 </section>
31 <!-- Image element - set the background image for the header in the line below -->
32 <div class="py-5 bg-image-full" style="background-image: url('https://source.unsplash.com/4ulffa6qoKA/1200x800')">
33   <!-- Put anything you want here! The spacer below with inline CSS is just for demo purposes! -->
34   <div style="height: 20rem"></div>
35 </div>
```

```
← → Search
author_detail.html author_list.html book_detail.html book_list.html base_generic.html # styles.css 4 index.html x ...
acts > myworld > locallibrary > locallibrary > catalog > templates > index.html > section.py-5 > div.container.my-5 > div.row.justify-content-center > div.col-lg-6 > p.lead
17 <div class="col-lg-6">
18   <h2>Library Record Counts</h2>
19   <p>The Library has the following record counts:</p>
20   <ul>
21     <li><strong>Books:</strong> {{ num_books }}</li>
22     <li><strong>Copies:</strong> {{ num_instances }}</li>
23     <li><strong>Copies available:</strong> {{ num_instances_available }}</li>
24     <li><strong>Authors:</strong> {{ num_authors }}</li>
25     <li><strong>Language:</strong> {{ num_language }}</li>
26   </ul>
27 </div>
28 </div>
29 </div>
30 </section>
31 <!-- Image element - set the background image for the header in the line below -->
32 <div class="py-5 bg-image-full" style="background-image: url('https://source.unsplash.com/4ulffa6qoKA/1200x800')">
33   <!-- Put anything you want here! The spacer below with inline CSS is just for demo purposes! -->
34   <div style="height: 20rem"></div>
35 </div>
36 <!-- Content section -->
37 <section class="py-5">
38   <div class="container my-5">
39     <div class="row justify-content-center">
40       <div class="col-lg-6">
41         <h2>Library</h2>
42         <p class="lead">A Masterpiece Made By TingYu Chou.</p>
43       </div>
44     </div>
45   </div>
46 </section>
47 <!-- Footer -->
48 <footer class="py-5 bg-dark">
49   <div class="container"><p class="m=0 text-center text-white">Copyright &copy; TingYu's Website 2022</p></div>
50 </footer>
51 {% endblock %}
```

• 美化後



Library Record Counts

The library has the following record counts:

- Books: 4
- Copies: 4
- Copies available: 2
- Authors: 3
- Language: 5







## Library

A Masterpiece Made By TingYu Chou.

Copyright © TingYu's Website 2022

Local Library

Home All books All authors

**Title: UNNATURAL**  
Author: [Kim HyunJung](#)  
Summary: WJSN - UNNATURAL (9TH MINI ALBUM)  
ISBN: 0000020210408  
Language: English  
Genre: Sci-Fi

### Preview



Local Library

Home All books All authors

Author: Kim, HyunJung



Kim, HyunJung (Dec. 24, 1994 - Dec. 24, 2094)

- Books
- [UNNATURAL](#) (1)  
WJSN - UNNATURAL (9TH MINI ALBUM)
  - [NEVERLAND](#) (1)  
WJSN - NEVERLAND (MINI ALBUM)