Books.py admin

from django.contrib import admin  
  
# Register your models here.  
from books.models import Book  
admin.site.register(Book)

models.py

from django.db import models  
  
# Create your models here.  
class Book(models.Model):  
 # Attributes #table definition  
 title=models.CharField(max\_length=20)  
 author= models.CharField(max\_length=20)  
 pages=models.IntegerField()  
 price=models.IntegerField()  
 language= models.CharField(max\_length=20)

urls.py

from django.contrib import admin  
from django.urls import path  
from books import views  
app\_name="books"  
  
urlpatterns = [  
 path('',views.home,name="home"),  
 path('add',views.add\_books,name="add"),  
 path('view',views.view\_books,name="views"),  
  
]

views.py

def home(request):  
 context={'name':'Arun','age':27}  
 return render(request,'home.html',context)  
  
def add\_books(request):  
 return render(request,'add.html')  
  
def view\_books(request):  
 return render(request,'view.html')

2nd library setting.py

*"""  
Django settings for library project.  
  
Generated by 'django-admin startproject' using Django 5.1.  
  
For more information on this file, see  
https://docs.djangoproject.com/en/5.1/topics/settings/  
  
For the full list of settings and their values, see  
https://docs.djangoproject.com/en/5.1/ref/settings/  
"""*from pathlib import Path  
import os  
  
# Build paths inside the project like this: BASE\_DIR / 'subdir'.  
BASE\_DIR = Path(\_\_file\_\_).resolve().parent.parent  
  
  
# Quick-start development settings - unsuitable for production  
# See https://docs.djangoproject.com/en/5.1/howto/deployment/checklist/  
  
# SECURITY WARNING: keep the secret key used in production secret!  
SECRET\_KEY = 'django-insecure-f\*(3hp%94yf#c)o&o30vx!!j^vtdi8@\_4\_1@2oq-kv&4)%#p$l'  
  
# SECURITY WARNING: don't run with debug turned on in production!  
DEBUG = True  
  
ALLOWED\_HOSTS = []  
  
  
# Application definition  
  
INSTALLED\_APPS = [  
 'django.contrib.admin',  
 'django.contrib.auth',  
 'django.contrib.contenttypes',  
 'django.contrib.sessions',  
 'django.contrib.messages',  
 'django.contrib.staticfiles','books','users',  
]  
  
MIDDLEWARE = [  
 'django.middleware.security.SecurityMiddleware',  
 'django.contrib.sessions.middleware.SessionMiddleware',  
 'django.middleware.common.CommonMiddleware',  
 'django.middleware.csrf.CsrfViewMiddleware',  
 'django.contrib.auth.middleware.AuthenticationMiddleware',  
 'django.contrib.messages.middleware.MessageMiddleware',  
 'django.middleware.clickjacking.XFrameOptionsMiddleware',  
]  
  
ROOT\_URLCONF = 'library.urls'  
  
TEMPLATES = [  
 {  
 'BACKEND': 'django.template.backends.django.DjangoTemplates',  
 'DIRS': [os.path.join(BASE\_DIR,'template')],  
 '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',  
 ],  
 },  
 },  
]  
STATICFILES\_DIRS=[os.path.join(BASE\_DIR,'static')]  
  
WSGI\_APPLICATION = 'library.wsgi.application'  
  
  
# Database  
# https://docs.djangoproject.com/en/5.1/ref/settings/#databases  
  
DATABASES = {  
 'default': {  
 'ENGINE': 'django.db.backends.sqlite3',  
 'NAME': BASE\_DIR / 'db.sqlite3',  
 }  
}  
  
  
# Password validation  
# https://docs.djangoproject.com/en/5.1/ref/settings/#auth-password-validators  
  
AUTH\_PASSWORD\_VALIDATORS = [  
 {  
 'NAME': 'django.contrib.auth.password\_validation.UserAttributeSimilarityValidator',  
 },  
 {  
 'NAME': 'django.contrib.auth.password\_validation.MinimumLengthValidator',  
 },  
 {  
 'NAME': 'django.contrib.auth.password\_validation.CommonPasswordValidator',  
 },  
 {  
 'NAME': 'django.contrib.auth.password\_validation.NumericPasswordValidator',  
 },  
]  
  
  
# Internationalization  
# https://docs.djangoproject.com/en/5.1/topics/i18n/  
  
LANGUAGE\_CODE = 'en-us'  
  
TIME\_ZONE = 'UTC'  
  
USE\_I18N = True  
  
USE\_TZ = True  
  
  
# Static files (CSS, JavaScript, Images)  
# https://docs.djangoproject.com/en/5.1/howto/static-files/  
  
STATIC\_URL = 'static/'  
  
# Default primary key field type  
# https://docs.djangoproject.com/en/5.1/ref/settings/#default-auto-field  
  
DEFAULT\_AUTO\_FIELD = 'django.db.models.BigAutoField'

Urls.py

from django.contrib import admin  
from django.urls import path,include  
  
urlpatterns = [  
 path('admin/', admin.site.urls),  
 path('',include('books.urls')),  
 path('users/',include('users.urls')),  
]

static-🡪css🡪style.css

template

add.html

<!DOCTYPE html>  
{% extends 'base.html' %}  
{% load static %}  
  
{% block content %}  
<h2>Add Books</h2>  
{% endblock %}

Base.html

<!DOCTYPE html>  
<html lang="en">  
{% load static %}  
<head>  
 <meta charset="UTF-8">  
 <title>Title</title>  
 <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC" crossorigin="anonymous">  
 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js" integrity="sha384-MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/tWtIaxVXM" crossorigin="anonymous"></script>  
 <link rel="stylesheet" href="{% static 'css/style.css' %}">  
</head>  
<body>  
<nav class="navbar navbar-expand-lg navbar-light bg-info">  
 <div class="container">  
 <a class="navbar-brand" href="#">LibraryApp</a>  
 <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">  
 <span class="navbar-toggler-icon"></span>  
 </button>  
 <div class="collapse navbar-collapse" id="navbarSupportedContent">  
 <ul class="navbar-nav ms-auto mb-2 mb-lg-0">  
 <li class="nav-item">  
 <a class="nav-link" href="{% url 'books:home' %}">Home</a>  
 </li>  
 <li class="nav-item">  
 <a class="nav-link" href="{% url 'books:add' %}">Add Books</a>  
 </li>  
 <li class="nav-item">  
 <a class="nav-link" href="{% url 'books:views' %}">View Books</a>  
 </li>  
 <li class="nav-item">  
 <a class="nav-link" href="{% url 'users:register' %}">Register</a>  
 </li>  
 <li class="nav-item">  
 <a class="nav-link" href="{% url 'users:login' %}">Login</a>  
 </li>  
 <li class="nav-item">  
 <a class="nav-link" href="{% url 'users:logout' %}">Logout</a>  
 </li>  
 </ul>  
  
 <form class="d-flex" style="margin-left:200px;">  
 <input class="form-control me-2" type="search" placeholder="Search" aria-label="Search">  
 <button class="btn btn-outline-primary" type="submit">Search</button>  
 </form>  
 </div>  
  
 </div>  
</nav>  
{% block content %}  
{% endblock %}  
</body>  
</html>

Home.html

<!DOCTYPE html>  
{% extends 'base.html' %}  
{% load static %}  
  
{% block content %}  
<h2 class="text-center text-info mt-3 ">Welcome to Library Application</h2>  
<h2>Name is {{name}}</h2>  
<h2>Age is {{age}}</h2>  
{% endblock %}  
<!--<a href="{% url 'books:home' %}">Home</a>-->  
<!--<a href="{% url 'books:add' %}">Add Books</a>-->  
<!--<a href="{% url 'books:views' %}">View Books</a>-->  
<!--<a href="{% url 'users:register' %}">Register</a>-->  
<!--<a href="{% url 'users:login' %}">Login</a>-->  
<!--<a href="{% url 'users:logout' %}">Logout</a>-->

Login.html

<!DOCTYPE html>  
{% extends 'base.html' %}  
{% load static %}  
  
{% block content %}  
<h2>Login</h2>  
{% endblock %}

Register.html

<!DOCTYPE html>  
{% extends 'base.html' %}  
{% load static %}  
  
{% block content %}  
<h2>Register</h2>  
{% endblock %}

View.html

<!DOCTYPE html>  
{% extends 'base.html' %}  
{% load static %}  
  
{% block content %}  
<h2>View Books</h2>  
{% endblock %}

Uers-🡪urls.py

from django.contrib import admin  
from django.urls import path  
from users import views  
app\_name="users"  
  
urlpatterns = [  
 path('login',views.login,name='login'),  
 path('reg',views.register,name='register'),  
 path('logout',views.logout,name='logout'),  
]

view.py

from django.shortcuts import render  
  
# Create your views here.  
def register(request):  
 return render(request,'register.html')  
  
def login(request):  
 return render(request,'login.html')  
  
def logout(request):  
 return login(request)