

<https://github.com/nikolovanela/dnick>

Settings.py

"""

Django settings for books project.

Generated by 'django-admin startproject' using Django 5.2.1.

For more information on this file, see

<https://docs.djangoproject.com/en/5.2/topics/settings/>

For the full list of settings and their values, see

<https://docs.djangoproject.com/en/5.2/ref/settings/>

"""

from pathlib import Path

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/5.2/howto/deployment/checklist/>

SECURITY WARNING: keep the secret key used in production secret!

SECRET_KEY = 'django-insecure-

ocntryep7j&\$d4e#z3(rmw5a\$8yg1aihyu&98i5^fkmwu38+47'

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',  
    'booksApp'  
]
```

```
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 = 'books.urls'
```

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

```
WSGI_APPLICATION = 'books.wsgi.application'
```

```
# Database
```

```
# https://docs.djangoproject.com/en/5.2/ref/settings/#databases
```

```
DATABASES = {
```

```
    'default': {
```

```
        'ENGINE': 'django.db.backends.sqlite3',
```

```
        'NAME': BASE_DIR / 'db.sqlite3',
```

```
    }
```

```
}
```

```
# Password validation
```

```
# https://docs.djangoproject.com/en/5.2/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/5.2/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/5.2/howto/static-files/
```

```
STATIC_URL = 'static/'
```

```
STATICFILES_DIRS = [BASE_DIR / 'static']
```

```
MEDIA_URL = '/media/'
```

```
MEDIA_ROOT = BASE_DIR / 'media'
```

```
# Default primary key field type
```

```
# https://docs.djangoproject.com/en/5.2/ref/settings/#default-auto-field
```

```
DEFAULT_AUTO_FIELD = 'django.db.models.BigAutoField'
```

```
Urls.py
```

```
from django.contrib import admin
```

```
from django.urls import path
```

```
from django.conf import settings
```

```
from django.conf.urls.static import static
```

```
from booksApp import views
```

```
urlpatterns = [
```

```
    path('admin/', admin.site.urls),
```

```
    path('', views.index, name='index'),
```

```
    path('index/', views.index, name='index'),
```

```
    path('book/<int:book_id>/', views.book_details, name='book_details'),
```

```
    path('book/addBook/', views.addBook, name='add_book'),
```

```
]
```

```
if settings.DEBUG:
```

```
    urlpatterns += static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)
```

```
admin.py
```

```
from django.contrib import admin
```

```
from booksApp.models import Book, Publisher
```

```
# Register your models here.
```

```
class BookAdmin(admin.ModelAdmin):
```

```
    pass
```

```
class PublisherAdmin(admin.ModelAdmin):
```

```
    pass
```

```
admin.site.register(Book, BookAdmin)
```

```
admin.site.register(Publisher, PublisherAdmin)
```

```
forms.py
```

```
from django import forms
```

```
from .models import Book
```

```
class BookForm(forms.ModelForm):
```

```
    def __init__(self, *args, **kwargs):
```

```
        super(BookForm, self).__init__(*args, **kwargs)
```

```
        for field in self.visible_fields():
```

```
            field.field.widget.attrs['class'] = 'form-control'
```

```
class Meta:
    model = Book
    fields = '__all__'
```

models.py

```
from django.db import models
```

```
# Create your models here.
```

```
class Publisher(models.Model):
    name = models.CharField(max_length=200, null=True, blank=True)
    country = models.CharField(max_length=200, null=True, blank=True)
    city = models.CharField(max_length=200, null=True, blank=True)
    year = models.IntegerField(null=True, blank=True)
    website = models.URLField(null=True, blank=True)

    def __str__(self):
        return self.name
```

```
class Book(models.Model):
    TYPE_CHOICES = [
        ('M', 'meka'),
        ('T', 'tvrda')
    ]
```

```
CATEGORY_CHOICES = [
    ('R', 'romansa'),
    ('T', 'triler'),
    ('B', 'biografija'),
    ('K', 'klasik'),
    ('D', 'drama'),
    ('I', 'istorija')
```

```
]
```

```
title = models.CharField(max_length=200, null=True, blank=True)
image = models.ImageField(upload_to='images/', null=True, blank=True,
default='images/Book_1.png')
isbn = models.CharField(max_length=50, null=True, blank=True)
year_of_publishing = models.IntegerField(null=True, blank=True)
publisher = models.ForeignKey(Publisher, on_delete=models.CASCADE, null=True,
blank=True)
pages = models.IntegerField(null=True, blank=True)
dimensions = models.CharField(max_length=50, null=True, blank=True)
type = models.CharField(max_length=1, choices=TYPE_CHOICES, null=True, blank=True)
category = models.CharField(max_length=1, choices=CATEGORY_CHOICES, null=True,
blank=True)
price = models.IntegerField(null=True, blank=True)

def __str__(self):
    return self.title
```

views.py

```
from django.shortcuts import render, redirect
from booksApp.models import Book
from .forms import BookForm

# Create your views here.

def index(request):
    books = Book.objects.all()
    return render(request, 'index.html', {'books': books})

def book_details(request, book_id):
    book = Book.objects.get(pk=book_id)
```

```
return render(request, 'bookDetails.html', {'book':book})
```

```
def addBook(request):
    if request.method == 'POST':
        form = BookForm(request.POST, request.FILES)
        if form.is_valid():
            form.save()
            return redirect('index')
    else:
        form = BookForm()
    return render(request, 'addBook.html', {'form': form})
```

addbook.html

```
{% extends 'base.html' %}
```

```
{% block content %}
```

```
<div class="container-fluid bg-danger">
    <h3 class="text-center"> Dodaj nova kniga</h3>
</div>
```

```
<div>
```

```
    <form method="post" action="{% url 'add_book' %}" enctype="multipart/form-data"
class="mb-4">
```

```
        {% csrf_token %}
```

```
        {{form.as_p}}
```

```
        <button class="btn btn-danger">Додај книга</button>
```

```
    </form>
```

```
</div>
```

```
{% endblock %}
```

Base.html


```

{% load static %}
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Title</title>
  <link rel="stylesheet" href="{% static 'css/bootstrap.css' %}" />
  <script src="{% static 'js/bootstrap.bundle.js' %}"></script>
</head>
<body>

<div class="container-fluid bg-primary h-50">
  <div class="d-flex justify-content-center">
    
  </div>
</div>

{% block content %}{% endblock %}

</body>
</html>

```

bookDetails.html

```

{% extends 'base.html' %}

{% block content %}
  <div class="container-fluid bg-danger">
    <h3 class="text-center"> {{ book.title }}</h3>
  </div>

  <div class="container-fluid my-5">
    <div class="row">
      <div class="col-6 " style="padding-left: 100px">
        
      </div>
    </div>
  </div>

```

</div>

<div class="col-6">

<p>Издавачка куќа: {{ book.publisher.name }}</p>

<p>Категорија: {{ book.category }}</p>

<p>ISBN: {{ book.isbn }}</p>

<p>Година на издавање: {{ book.year_of_publishing }}</p>

<p>Број на страни: {{ book.pages }}</p>

<p>Димензии: {{ book.dimensions }}</p>

<p>Корица: {{ book.type }}</p>

<p>Цена: {{ book.price }}</p>

</div>

</div>

</div>

Nazad kon pocetna
{% endblock %}

Index.html

{% extends 'base.html' %}

{% load static %}

{% block content %}

<div class="container-fluid bg-danger h-25">

<p class="text-danger">gegeg</p>

</div>

<div class="container-fluid ">

<div class="d-flex justify-content-end">

Dodaj nova
kniga

</div>

<div class="row">

{% for book in books %}

```
<div class="col-4">
  <a href="{% url 'book_details' book.id %}" style="color: #cc6464">
    <div class="card" style="width: 18rem;">
      {% if book.image %}
        
      {% else %}
        
      {% endif %}
      <div class="card-body">
        <h5 class="card-title">{{ book.title }}</h5>
        <p class="card-text">Цена: {{ book.price }}</p>
      </div>
    </div>
  </a>
</div>
{% endfor %}
</div>
{% endblock %}
```