# Министерство образования и науки Российской Федерации Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования «Московский государственный технический университет имени Н.Э. Баумана» (МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ	«Информатики и систем управления»
КАФЕДРА	«Систем обработки информации и управления»

# Домашнее задание

Проектирование веб-приложения с использованием
Фреймворка Django.

ПО ДИСЦИПЛИНЕ: «Разработка интернет приложений»

Студент ИУ5-52Б  $(\Gamma pynna)$ 

<u>Чепкин Д. А.</u> (И.О. Фамилия)

## Описание задания

Создайте прототип веб-приложения с использованием фреймворка Django на основе базы данных, реализующий концепцию master/detail. Прототип должен содержать:

- 1. Две модели, связанные отношением один-ко-многим.
- 2. Стандартное средство администрирования Django позволяет редактировать данные моделей. Желательно настроить русификацию ввода и редактирования данных.
- 3. Веб-приложение формирует отчет в виде отдельного view/template, отчет выводит HTML-страницу, содержащую связанные данные из двух моделей.
- 4. Для верстки шаблонов используется фреймворк Bootstrap, или аналогичный фрейворк по желанию студента.

#### Модели:

```
from django.db import models
from django.shortcuts import reverse
# Create your models here.
class Post(models.Model):
   title = models.CharField(max_length=150, db_index=True)
   slug = models.SlugField(max_length=150, unique=True)
   author= models.CharField(max_length=150, db_index=True)
   body = models.TextField(blank=True, db_index=True)
   tags = models.ManyToManyField('Tag', blank=True, related_name='posts')
   date pub = models.DateTimeField(auto now add=True)
   user = models.ForeignKey('User', on_delete=models.CASCADE)
   def get_absolute_url(self):
       return reverse('post_detail_url', kwargs={'slug': self.slug})
   def __str__(self):
       return self.title
class Tag(models.Model):
   title = models.CharField(max_length=50)
   slug = models.SlugField(max_length=50, unique=True)
   def get_absolute_url(self):
       return reverse('tag_detail_url', kwargs={'slug': self.slug})
   def __str__(self):
       return '{}'.format(self.title)
```

```
class User(models.Model):
    name = models.CharField(max_length=50)
    password = models.CharField(max_length=50)

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

#### views.py:

```
from django.shortcuts import render
from .models import *
# Create your views here.
def posts_list(request):
    posts = Post.objects.all()
    return render(request, 'blog/index.html', context={'posts': posts})
def post_detail(request, slug):
    post = Post.objects.get(slug_iexact=slug)
    return render(request, 'blog/post detail.html', context={'post': post})
def tags_list(request):
    tags = Tag.objects.all()
    return render(request, 'blog/tags list.html', context={'tags': tags})
def tag_detail(request, slug):
    tag = Tag.objects.get(slug__iexact=slug)
    return render(request, 'blog/tag_detail.html', context={'tag': tag})
def main(request):
   return render(request, 'blog/main.html')
```

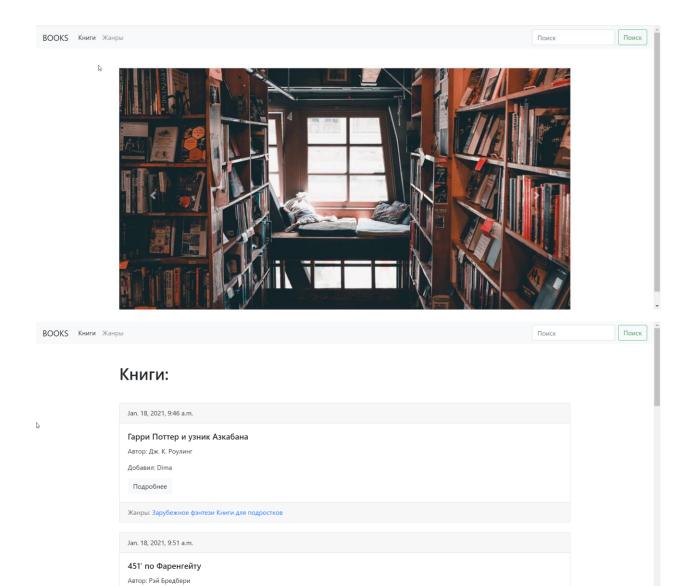
#### Описание:

Макет сайта содержит макет базы данных с книгами, жанрами и администраторами.

Книги и жанры связаны соотношением многие-ко-многим

Книги и администраторы связаны соотношением один-ко многим

Скриншоты сайта:



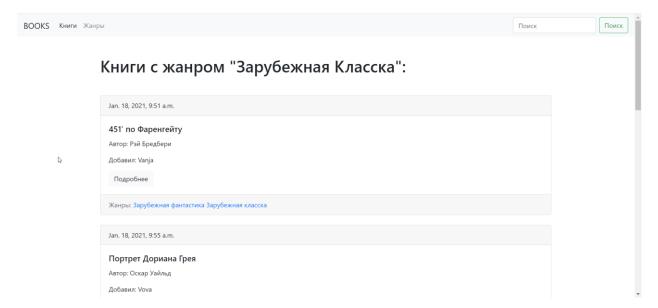
Поиск

# Жанры:

Добавил: Vanja

BOOKS Книги Жанры

Зарубежное фэнтези
Книги для подростков
Зарубежная фантастика
Зарубежная класска
Классическая литература
Классическая проза



### Форма администрирования Django:

