

Московский государственный технический университет им. Н.Э. Баумана  
Факультет «Информатика и системы управления»

ДИСЦИПЛИНА:  
«Разработка интернет-приложений»

Отчет по лабораторной работе №4  
«Django. Шаблонизация»

Выполнил:  
Студент 3 курса  
Факультет ИУ  
Группа ИУ5-51Б  
Ерохин И.А.  
Преподаватель:  
Гапанюк Ю.Е.

Москва, 2019 г.

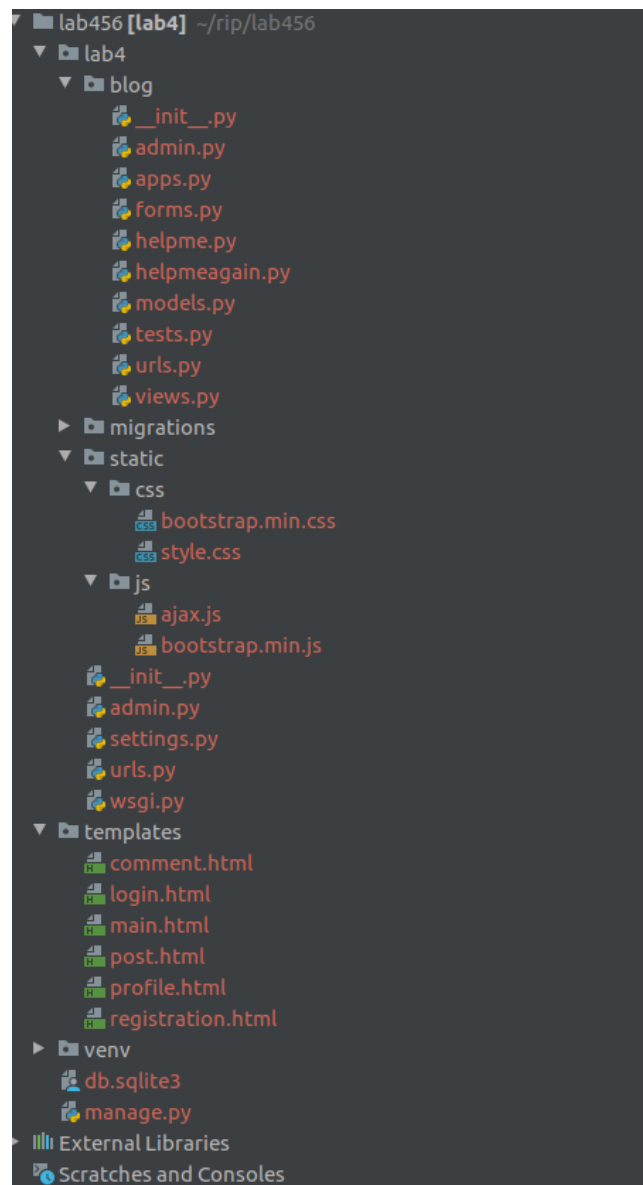
## 1. Задание

В этой ЛР вы создадите Django-проект, покажете пользователю статичную страницу, познакомитесь с конструкциями шаблонизаторов: переменные, теги, наследование шаблонов.

- ☐ Создать проект
- ☐ Реализовать view, в которых генерируются html-страницы
- ☐ В шаблонах должны быть использованы рассмотренные конструкции: переменные, вложенные значения, циклы, условия
- ☐ Все шаблоны должны расширять базовый шаблон
- ☐ Для элементов списка использовать тег `include`
- ☐ По нажатии на элемент списка должна открываться страница информации об элементе
- ☐ Для верстки необходимо использовать Bootstrap

## 2. Выполнение задания и исходный код проекта

Структура проекта:



models.py:

```
from django.db import models

class Post(models.Model):
    post_head = models.CharField(max_length=50)
    post_text = models.CharField(max_length=255)
    publication_date = models.DateField('Date published')
    def __str__(self):
        return self.post_head

class Comment(models.Model):
    id_post = models.ForeignKey(Post, on_delete=models.CASCADE)
    answer_text = models.CharField(max_length=70)
    rating = models.IntegerField(default=0)
    def __str__(self):
        return self.answer_text
```

views.py:

```
from django.shortcuts import render, get_object_or_404
from django.db.models import Count
from django.views import generic
from .models import *

class MainView(generic.ListView):
    template_name = 'main.html'
    context_object_name = 'main_postList'

    def get_queryset(self):
        return Post.objects.annotate(num=Count('comment')).order_by('-publication_date')[:5]

class PostView(generic.DeleteView):
    model = Post
    template_name = 'post.html'
```

Использование наследования шаблонов, тегов и переменных (шаблон main.html):

```
<div class="justify-content-center w-50 mx-auto">
    {% block body %}

    {% if main_postList %}
        {% for post in main_postList %}
            <div class="card m-2 mb-4 mt-4">
                <div class="card-body">
                    <h5 class="card-title">
                        <a href="/{{ post.id }}">{{ post.post_head }}</a>
                    </h5>
                    <p class="card-text">{{ post.post_text }}</p>
                </div>
                <div class="card-footer">
                    <span><i class="fas fa-comments"></i> {{ post.num }}</span>
                </div>
            </div>
        {% endfor %}
    {% else %}
```

```

        <div class="alert alert-info" role="alert">
            Доступных записей нет.
        </div>
    {% endif %}

    {% endblock %}
</div>

```

Подключение статических файлов:

```

{% load staticfiles %}
<link rel="stylesheet" href="{% static 'css/bootstrap.min.css' %}" type="text/css">
<script type="text/javascript"
src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.0/jquery.min.js"></script>
<script src="{% static 'js/bootstrap.min.js' %}"></script>

```

Использование include:

Шаблон post.html использует шаблон comment.html

```

<div class="card mt-4">
    <div class="card-body">
        <ul class="list-unstyled">
            {% for comment in post.comment_set.all %}
                {% include 'comment.html' with comment=comment %}
            {% empty %}
                <div class="alert alert-info" role="alert">
                    Комментариев к этой записи пока нет.
                </div>
            {% endfor %}
        </ul>
    </div>
</div>

```