|  |  |
| --- | --- |
|  | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ \_\_\_\_\_\_\_\_ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

КАФЕДРА \_\_\_\_\_\_\_СИСТЕМЫ ОБРАБОТКИ ИНФОРМАЦИИ И УПРАВЛЕНИЯ (ИУ5)\_\_\_\_

**О Т Ч Е Т**

**по лабораторной работе**

по дисциплине: Разработка интернет приложений

на тему:\_Шаблонизация \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Студент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

(Группа) (Подпись, дата) (И.О.Фамилия)

Руководитель **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

(Подпись, дата) (И.О.Фамилия)

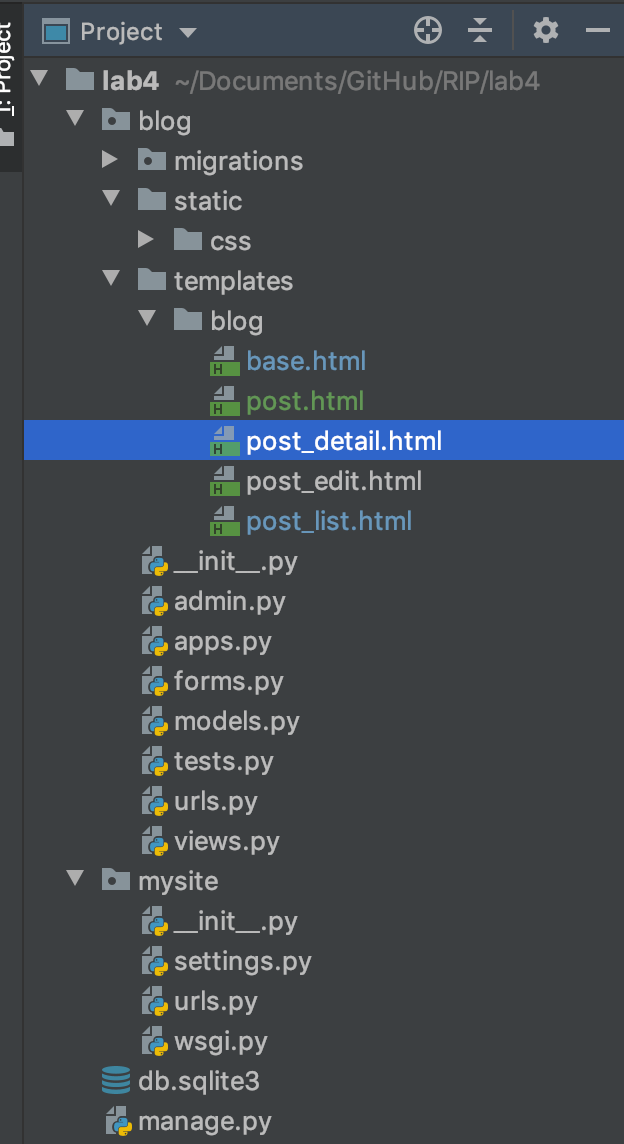
*2019 г.*

# Задание и порядок выполнения

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

### Структура проекта



### Главный шаблон (условия, переменные)

{% extends 'blog/base.html' %}

{% block content %}

<div class="post">

{% if post.published\_date %}

<div class="date">

{{ post.published\_date }}

</div>

{% endif %}

<a class="btn btn-default" href="{% url 'post\_edit' pk=post.pk %}"><span class="glyphicon glyphicon-pencil"></span></a>

<h1>{{ post.title }}</h1>

<p>{{ post.text|linebreaksbr }}</p>

</div>

{% endblock %}

### Включение шаблонов (использование циклов)

{% extends 'blog/base.html' %}

{% block content %}

{% for post in posts %}

<div class="post">

{%include 'blog/post.html' with post=post %}

</div>

{% endfor %}

{% endblock %}

### Urls.py

rom django.urls import path

from . import views

urlpatterns = [

path('', views.post\_list, name='post\_list'),

path('post/<int:pk>/', views.post\_detail, name='post\_detail'),

path('post/new/', views.post\_new, name='post\_new'),

path('post/<int:pk>/edit/', views.post\_edit, name='post\_edit'),

]

### Views.py

from django.shortcuts import render, redirect

from django.utils import timezone

from .models import Post

from django.shortcuts import render, get\_object\_or\_404

from blog.forms import PostForm

def post\_list(request):

posts = Post.objects.filter(published\_date\_\_lte=timezone.now()).order\_by('published\_date')

return render(request, 'blog/post\_list.html', {'posts': posts})

def post\_detail(request, pk):

post = get\_object\_or\_404(Post, pk=pk)

return render(request, 'blog/post\_detail.html', {'post': post})

def post\_new(request):

if request.method == "POST":

form = PostForm(request.POST)

if form.is\_valid():

post = form.save(commit=False)

post.author = request.user

post.published\_date = timezone.now()

post.save()

return redirect('post\_detail', pk=post.pk)

else:

form = PostForm()

return render(request, 'blog/post\_edit.html', {'form': form})

def post\_edit(request, pk):

post = get\_object\_or\_404(Post, pk=pk)

if request.method == "POST":

form = PostForm(request.POST, instance=post)

if form.is\_valid():

post = form.save(commit=False)

post.author = request.user

post.published\_date = timezone.now()

post.save()

return redirect('post\_detail', pk=post.pk)

else:

form = PostForm(instance=post)

return render(request, 'blog/post\_edit.html', {'form': form})

### 