

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



Отчет
Лабораторная работа № 4
По курсу «Разработка интернет-приложений»
«Шаблонизация»

ИСПОЛНИТЕЛЬ:

Матиенко Андрей

Группа ИУ5-51

" __ " _____ 2019 г.

ПРЕПОДАВАТЕЛЬ:

Гапанюк Ю.Е.

" __ " _____ 2019 г.

Москва 2019

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 нажатию на элемент списка должна открываться страница информации об элементе
- Для верстки необходимо использовать Bootstrap

Views.py:

```
manage.py x settings.py x views.py x example.html x models.py x
1 from django.shortcuts import render
2 from django.http import HttpResponse
3 from .models import Post
4 from django.utils import timezone
5 from django.shortcuts import get_object_or_404
6 from .forms import PostForm
7 from django.shortcuts import redirect
8 # Create your views here.
9 #def index(request):
10 #    return HttpResponse("Hello, World! I'm at the 'my_app' repository")
11
12 # def get_list(request):
13 #     my_variable = "WTF???"
14 #     return render(request, 'example.html', {'my_variable': my_variable})
15
16 def get_list(request):
17     posts = Post.objects.filter(published_date__lte=timezone.now()).order_by('published_date')
18     return render(request, 'example.html', {'posts': posts})
19
20 def post_detail(request, pk):
21     Post.objects.get(pk=pk)
22     post = get_object_or_404(Post, pk=pk)
23     return render(request, 'post_detail.html', {'post': post})
24
25 def post_new(request):
26     if request.method == "POST":
27         form = PostForm(request.POST)
28         if form.is_valid():
29             post = form.save(commit=False)
30             post.author = request.user
31             post.published_date = timezone.now()
32             post.save()
```

Base.html:

```
Project
├── lab_4
│   ├── migrations
│   ├── static
│   └── templates
│       ├── base.html
│       ├── example.html
│       ├── post_detail.html
│       ├── post_edit.html
│       ├── _init_.py
│       ├── admin.py
│       ├── apps.py
│       ├── forms.py
│       ├── models.py
│       ├── tests.py
│       ├── urls.py
│       └── views.py
├── mysite
│   ├── _init_.py
│   ├── settings.py
│   ├── urls.py
│   ├── wsgi.py
│   ├── db.sqlite3
│   ├── get-pip.py
│   ├── manage.py
│   └── External Libraries
└── Scratches and Consoles

manage.py settings.py views.py example.html base.html models.py
1 <!DOCTYPE html>
2 {% load static %}
3 <html lang="en">
4 <head>
5 <link rel="stylesheet" href="//maxcdn.bootstrapcdn.com/bootstrap/3.2.0/css/bootstrap.min.css">
6 <link rel="stylesheet" href="//maxcdn.bootstrapcdn.com/bootstrap/3.2.0/css/bootstrap-theme.min.css">
7 <link rel="stylesheet" href="{% static 'css/blog.css' %}">
8 <link href="https://fonts.googleapis.com/css?family=Long+Cang&display=swap" rel="stylesheet">
9 <title>Andrew's site</title>
10 </head>
11 <body>
12 <div class="page-header">
13     {% if user.is_authenticated %}
14     <a href="{% url 'post_new' %}" class="top-menu"><span class="glyphicon glyphicon-plus"></span></a>
15     {% endif %}
16     <h1><a href="/">Andrew's blog</a></h1>
17 </div>
18
19 <div class="Space">
20 
21 </div>
22
23 <div class="content container">
24 <div class="row">
25 <div class="col-md-8">
26     {% block content %}
27     {% endblock %}
28 </div>
29 </div>
30 </body>
```

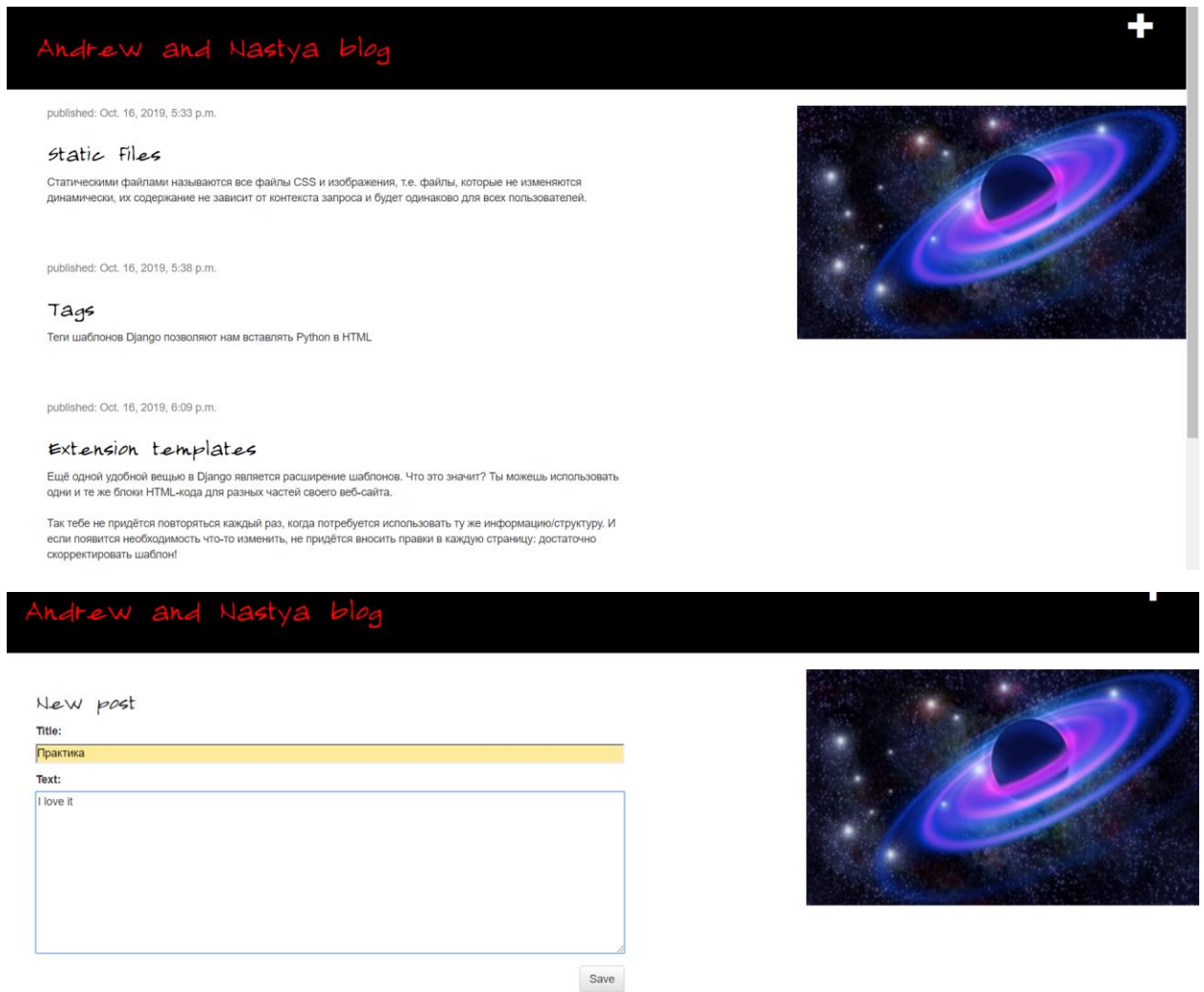
Шаблонизация:

```
manage.py settings.py views.py example.html base.html models.py
1 {% extends 'base.html' %}
2
3 {% block content %}
4     {% for post in posts %}
5         <div class="post">
6             <div class="date">
7                 <p>published: {{ post.published_date }}</p>
8             </div>
9             <h1><a href="{% url 'post_detail' pk=post.pk %}">{{ post.title }}</a></h1>
10            <p>{{ post.text|linebreaksbr }}</p>
11        </div>
12    {% endfor %}
13 {% endblock %}

manage.py settings.py views.py example.html post_detail.html base.html models.py
1 {% extends 'base.html' %}
2
3 {% block content %}
4     <div class="post">
5         {% if post.published_date %}
6             <div class="date">
7                 {{ post.published_date }}
8             </div>
9         {% endif %}
10        {% if user.is_authenticated %}
11            <a class="btn btn-default" href="{% url 'post_edit' pk=post.pk %}"><span class="glyphicon glyphicon-pencil"></span></a>
12        {% endif %}
13        <h1>{{ post.title }}</h1>
14        <p>{{ post.text|linebreaksbr }}</p>
15    </div>
16 {% endblock %}
```

В конечном итоге получаем во такой сайт:

Это блог, в который можно добавлять, редактировать и удалять записи:



Выводы:

Научились работе в Django. Узнали как верстаются сайты и познакомились на практике с данной работой.