

**Московский государственный
технический университет
им. Н.Э. Баумана**

**Разработка интернет-приложений
Лабораторная работа № 5
“Шаблонизация”**

Выполнил:
студент группы ИУ5-53
Степанов Д. Г.
Подпись:
Дата:

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

▼ lab5 C:\GIT\python_labs\lab5

▼ lab5

__init__.py

settings.py

urls.py

wsgi.py

▼ my_app

migrations

__init__.py

admin.py

apps.py

models.py

tests.py

urls.py

views.py

▼ static

favicon.png

style.css

▼ templates

▼ my_app

base.html

basextend.html

inclpage.html

index.html

page.html

db.sqlite3

lab5 review_docx.docx

manage.py

~\$b5 review_docx.docx

ЛР5. Шаблонизация.pdf

Ф а й л views.py

```
from django.shortcuts import render
from django.http import HttpRequest, HttpResponse
import datetime
from django.views import View

def basepage(request):
    return render(request, "my_app/base.html")

def basextend(request):
    colors = {"black": "#000000", "white": "#FFFFFF", "red": "#FF0000", "green": "#00FF00", "blue": "#0000FF"}
    return render(request, "my_app/basextend.html", colors)

def testpage(request):
    pages = {'pages': [
        {'title': "Первая страница", "id": 1},
        {'title': "Вторая страница", "id": 2},
        {'title': "Третья страница", "id": 3}
    ]}
    server_time = datetime.datetime.now()
    return render(request, "my_app/index.html", pages)

def function_view(request):
    return HttpResponse("Response from function view")

class ExampleClassBased(View):
    def get(self, request):
        return HttpResponse("response from class based view")

class PageView(View):
    def get(self, request, id):
        data = {'page': {'id': id}}
        return render(request, 'my_app/page.html', data)

class IncludePage(View):
    def get(self, request):
        return render(request, 'my_app/inclpage.html')
```

Ф а й л urls.py

```
from django.conf.urls import url, include
from django.contrib import admin
from my_app.views import function_view, testpage, ExampleClassBased, basepage, basextend, PageView, IncludePage

urlpatterns = [
    url(r'^basextend/', basextend, name='basextend'),
    url(r'^basepage/', basepage, name='basepage'),
    url(r'^testpage/', testpage, name='testpage'),
    url(r'^function_view/', function_view, name='function_view'),
    url(r'^class_based_view/', ExampleClassBased.as_view(), name='class_based_view'),
    url(r'^page/(?P<id>\d+)', PageView.as_view(), name='page'),
    url(r'^inclpage/', IncludePage.as_view(), name='includepage'),
    url(r'^$', testpage, name='testpage')
```

Б а з о в ы й ш а б л о н base.html

```
views.py x urls.py x base.html x index.html x
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>{% block title %}{% endblock %}</title>
6   {% block head %}{% endblock %}
7 </head>
8 <body>
9   {% block content %}{% endblock %}
10  {% block footer %}{% endblock %}
11 </body>
12 </html>
```

Н а ч а л ь н а я с т р а н и ц а (п р и м е р и с п о л ь з о в а н и я т е г о в , н а с л е д о в а н и я , bootstrap, п о д к л ю ч е н и я static-ф а й л о в , include)

```
{% extends 'my_app/base.html' %}
{% load static %}

{% block title %} My_app {% endblock %}
{% block head %}
  <meta charset="UTF-8">
  <link rel = "icon" type = "image/png" href="{% static 'favicon.png' %}">
  <link rel = "stylesheet" href = "https://cdnjs.cloudflare.com/ajax/libs/twitter-bootstrap/
  <link rel = "stylesheet" href = "{% static '/style.css' %}">
{% endblock %}
{% block content %}
<div class = "container">
  <div class="row">
    <div class = "col-lg-4 col-lg-offset-4">
      <h2 class = "title">Начальная Страница</h2>
      <h2 class = "title">{{ server_time }}</h2>
      <ul>
        {% for page in pages %}
        <li><a href = "{% url 'page' page.id %}">{{ page.title }}</a></li>
        {% empty %}
        <li>Пустая ссылка</li>
        {% endfor %}
      </ul>
    </div>
  </div>
</div>
<ul>
  {% for page in pages %}
    {% include 'my_app/inclpage.html' with element=page %}
  {% empty %}
    <li>Пустой список</li>
  {% endfor %}
</ul>
<script src = "https://code.jquery.com/jquery-2.1.4.min.js"></script>
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js" integrity="
{% endblock %}
```

И с п о л ь з о в а н и е ц и к л о в

```
{% extends 'my_app/base.html' %}
{% load static %}
{% block title %}Extend{% endblock %}
{% block head %}<style> .txt{ color: {{ col.red }}; } </style>{% endblock %}
{% block content %}
    <h2>extended page from <i>my_app/base.html</i> </h2>
    <div class="txt">Цветная строка</div>
    <b>Список цветов:</b>
    <ul>
        {% for k, v in colors.items %}
            <li>{{ k }} = {{ v }}</li>
        {% endfor %}
    </ul>
{% endblock %}
```

Страница с include

```
views.py x urls.py x base.html x index.html x page.html x inclpage.html x basexte
1 {% extends 'my_app/base.html' %}
2 {% block title %}IncludePage {% endblock %}
3 {% block head %}{% endblock %}
4 {% block content %}
5     <li><a href = "{% url 'page' element.id %}">{{ element.title }} included page</a></li>
6 {% endblock %}
```

Результат выполнения программы

← → ↻ 127.0.0.1:8000

Начальная Страница

Nov. 3, 2017, 7:46 p.m.

- Первая страница
- Вторая страница
- Третья страница

- Первая страница included page
- Вторая страница included page
- Третья страница included page

← → ↻ 127.0.0.1:8000/class_based_view/

response from class based view

← → ↻ 127.0.0.1:8000/page/1

Page number 1



Response from fucntion view