Московский государственный университет им. Н. Э. Баумана

Факультет «Информатика и системы управления»

Кафедра «Системы обработки информации и управления»

Лабораторные работы по курсу:

«Разработка Интернет Приложений»

ЛР6. Шаблонизация

Исполнитель:

Студент группы РТ5-51

Чечнев А.А.

Преподаватель:

Гапанюк Ю. Е.

«___»___________________________

Москва 2017 г.

Задание и порядок выполнения

В этой ЛР необходимо создать Django-проект, показать пользователю статичную страницу, познакомиться 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

Исходный код:

settings.py:

```
import os

# Build paths inside the project like this: os.path.join(BASE_DIR, ...)
BASE_DIR = os.path.dirname(os.path.dirname(os.path.abspath(__file__)))


# Quick-start development settings - unsuitable for production
# See https://docs.djangoproject.com/en/1.11/howto/deployment/checklist/

# SECURITY WARNING: keep the secret key used in production secret!
SECRET_KEY = '&h2(i)xy$m^m1wcap@yfd+71l@c*q%ci116s2$76m=_bu(&76z'

# SECURITY WARNING: don't run with debug turned on in production!
DEBUG = True

ALLOWED_HOSTS = []


# Application definition

INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'myapp.apps.MyappConfig',

]

MIDDLEWARE = [
    'django.middleware.security.SecurityMiddleware',
    'django.contrib.sessions.middleware.SessionMiddleware',
    'django.middleware.common.CommonMiddleware',
    'django.middleware.csrf.CsrfViewMiddleware',
    'django.contrib.auth.middleware.AuthenticationMiddleware',
    'django.contrib.messages.middleware.MessageMiddleware',
    'django.middleware.clickjacking.XFrameOptionsMiddleware',
]

ROOT_URLCONF = 'untitled3.urls'

TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [os.path.join(BASE_DIR, 'templates')]
        ,
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [
                'django.template.context_processors.debug',
                'django.template.context_processors.request',
                'django.contrib.auth.context_processors.auth',
                'django.contrib.messages.context_processors.messages',
            ],
        },
    },
]

WSGI_APPLICATION = 'untitled3.wsgi.application'


# Database
# https://docs.djangoproject.com/en/1.11/ref/settings/#databases

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': os.path.join(BASE_DIR, 'db.sqlite3'),
    }
}


# Password validation
# https://docs.djangoproject.com/en/1.11/ref/settings/#auth-password-
validators

AUTH_PASSWORD_VALIDATORS = [
    {
        'NAME':
'django.contrib.auth.password_validation.UserAttributeSimilarityValidator',
    },
    {
        'NAME':
'django.contrib.auth.password_validation.MinimumLengthValidator',
    },
    {
        'NAME':
'django.contrib.auth.password_validation.CommonPasswordValidator',
    },
    {
        'NAME':
'django.contrib.auth.password_validation.NumericPasswordValidator',
    },
]


# Internationalization
# https://docs.djangoproject.com/en/1.11/topics/i18n/

LANGUAGE_CODE = 'en-us'
```

```
TIME_ZONE = 'UTC'

USE_I18N = True

USE_L10N = True

USE_TZ = True


# Static files (CSS, JavaScript, Images)
# https://docs.djangoproject.com/en/1.11/howto/static-files/

STATIC_URL = '/static/'
```

urls.py

```
from django.conf.urls import url, include
from django.contrib import admin
from myapp.views import *


urlpatterns = [
    url(r'^admin/', admin.site.urls),
    url(r'^main/', show_main, name="main_url"),
    url(r'^info/', show_info, name="info_url"),
    url(r'^goods/$', goods_view, name="goods_url"),
    url(r'^goods/([\d+]{1,4})/$', good_view1, name="good_numb_url"),
    url(r'^goods/([\d+]{1,4})/characteristic', good_char,
name="good_char_url"),
    url(r'^contacts/$', OrderView.as_view(), name="contacts_url")
]
```

views.py

```
from django.shortcuts import render
from django.views.generic import View
from django.http import HttpResponse


# Create your views here.
def show_base(request):
    title = 'Site'
    head_text = 'Hello'

    return render(request, "base.html", locals())


def show_main(request):
    title = 'Site'
    head_text = 'Main'
    main_text = "This is main prog"

    return render(request, "main.html", locals())


def show_info(request):
    title = 'SiteInfo'
    head_text = 'This is information'
    main_text = "Look! This is information"

    return render(request, "info.html", locals())


def goods_view(request):
    good_id = ""
    goods_list = "Goods_list"
    data = {'goodlist': []}
    for i in range(1, 10):
        data['goodlist'].append(
            {
                'id': i,
                'title': '{0}{1}'.format('Good ', i),
                'description': 'Very useful description №{}'.format(i),
                'text': 'Very useful text №{}'.format(i),
                'date': '29.12.2017'
            }
        )
    return render(request, "goods.html", locals())


def good_view(request):
    good_id = "Good site"
    # good_id = good_id1
    return render(request, "goods.html", locals())


def good_view1(request, id):
    good_id = id
    # good_id = good_id1
    return render(request, "goods.html", locals())


def good_char(request, id):
    good_id = id
    # good_id = good_id1
    q = request.GET["id"]
    return render(request, "goods.html", locals())


class OrderView(View):
    def get(self, request):
        q = self.kwargs.get("id")
        # q = request.GET.get("id")
        return render(request, 'order.html', locals())
```

main.css

```
@import url('https://fonts.googleapis.com/css?family=Lato|Raleway');
@import "bootstrap.css";

body{
    background-color: #f2f2f2;
    font-family: "Lato", serif;
    font-weight: 300;
    font-size: 16px;
    color: #555;

    -webkit-font-smoothing: antialiased;
    -webkit-overflow-scrolling: touch;

    margin-top: 70px;
}

h1, h2, h3, h4, h5, h6{
    font-family: "Raleway", serif;
    font-weight: 300;
    color: #2cff9d;
}

p{
    line-height: 28px;
    margin-bottom: 25px;
}

.centered{
    text-align: center;
}

a{
    color: #f85c37;
}

a:hover, a:focus{
    color: #7b7b7b;
    text-decoration: none;
}

hr{
    display: block;
    height: 1px;
    border: 0;
    border-top:1px solid #ccc;
    margin:1em 0;
    padding: 0;
}

.navbar{
    font-weight: 800;
    font-size: 14px;
    padding-top: 15px;
    padding-bottom: 15px;
}
.navbar-inverse{
    background: #2d0051;
    border-color: #6affe1;
    border-bottom-style: ridge;
}

.navbar-inverse .navbar.nav > .active > a{
    background-color: #ff7878;
}

.navbar-inverse .navbar-brand{
    color: #999;
    font-weight: bolder;
    font-size: 22px;
    letter-spacing: 1px;
}

{% load static %}

#headerwrap{
    background: url(office.jpg) no-repeat center top;
    margin-top: -10px;
    margin-bottom: 20px;
    text-align: center;
    backface-visibility: hidden;
    background-position: center center;
    min-height: 500px;
    width: 100%;

    background-size: 100%;

    -webkit-background-size: cover;
    -moz-background-size: cover;
```

base.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    {% load static %}
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>{{ title }}</title>
  <link rel="stylesheet" href={% static 'Bootstrap/css/bootstrap.min.css' %}>
  <link rel="stylesheet" href={% static 'Bootstrap/css/font-
awesome.min.css'%}>
  <link rel="stylesheet" href={% static 'Bootstrap/css/main.css'%}>
</head>
<body>
  <div class="navbar navbar-inverse navbar-fixed-top">
    <div class="container">
      <div class="navbar-header">
        <button type="button" class="navbar-toggle" data-toggle="collapse"
data-target=".navbar-collapse">
          <span class="icon-bar"></span>
          <span class="icon-bar"></span>
          <span class="icon-bar"></span>
        </button>
        <a class="navbar-brand" href="#"><i class="fa fa-ellipsis-h"></i>{{
head_text }}</a>
      </div>
      <div class="navbar-collapse collapse">
        <ul class="nav navbar-nav navbar-right">
            {% block head_buttons %}{% endblock %}

        </ul>
      </div>
    </div>
  </div>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.1.0/jquery.min.js"></scri
pt>
<script src={% static "Bootstrap/js/bootstrap.min.js" %}></script>
<div id = "headerwrap">
  <div class="container">
      <div class="row centered">
            <div class="col-lg-10 col-lg-offset-1">
                {%  block body1 %}{% endblock %}
            </div>
      </div>
    </div>
```

```
</div>


</body>

</html>
```

goods.html

```
{%  extends 'base.html' %}

{%  block head_buttons %}

      <li><a href={% url 'main_url' %}>Домой</a></li>
      <li><a href={% url "info_url" %}>Инфо</a></li>
      <li class="active"><a href={% url "goods_url"%}>Товары</a></li>
      <li><a href={% url "contacts_url" %}>Контакты</a></li>
      <li><a href={% url "good_numb_url" 1%}><i class="fa fa-envelope-o" aria-hidden="true"></i></a></li>

{% endblock %}


{% block body1 %}

    <h2>{{ goods_list }}</h2>

    {% if good_id != "" %}
        <h2>Good id is {{ good_id }}</h2>
        <a href={% url "good_char_url" good_id %}>Characteristic</a>

    {% else %}
                <li><a class="hh" href={% url "good_numb_url" 1%}>Good1</a></li>
                <li><a href={% url "good_numb_url" 2%}>Good2</a></li>
                <li><a href={% url "good_numb_url" 3%}>Good3</a></li>
                <li><a href={% url "good_numb_url" 4%}>Good4</a></li>
                <li><a href={% url "good_numb_url" 5%}>Good5</a></li>
                <li><a href={% url "good_numb_url" 6%}>Good6</a></li>
                <li><a href={% url "good_numb_url" 7%}>Good7</a></li>
    {% endif %}

{% endblock %}
```

info.html

```
{%  extends 'base.html' %}

{%  block head_buttons %}

      <li><a href={% url 'main_url' %}>Домой</a></li>
      <li class="active"><a href="#">Инфо</a></li>
      <li><a href={% url "goods_url" %}>Товары</a></li>
      <li><a href={% url "contacts_url" %}>Контакты</a></li>
      <li><a href="main1.html"><i class="fa fa-envelope-o" aria-hidden="true"></i></a></li>
```

```
{% endblock %}

{%  block body1 %}
    <h1>{{ main_text }}</h1>
{% endblock %}
```

main.html

```
{% extends "base.html" %}

{% block head_buttons %}
    <li class="active"><a href="#">Домой</a></li>
    <li><a href={% url 'info_url' %}>Инфо</a></li>
    <li><a href={% url "goods_url" %}>Товары</a></li>
    <li><a href={% url "contacts_url" %}>Контакты</a></li>
    <li><a href="#"><i class="fa fa-envelope-o" aria-
hidden="true"></i></a></li>
{% endblock %}

{% block body1 %}
    <div id="qwe">
        <h1> {{ main_text }} </h1>
    </div>
{% endblock %}
```

order.html

```
{% extends "base.html" %}

{% block body1 %} <h1>Class based view</h1>
    <h1>{{ q }}</h1>
{% endblock%}

{%  block head_buttons %}

      <li><a href={% url 'main_url' %}>Домой</a></li>
      <li><a href={% url "info_url" %}>Инфо</a></li>
      <li><a href={% url "goods_url"%}>Товары</a></li>
      <li class="active"><a href={% url "contacts_url" %}>Контакты</a></li>
      <li><a href={% url "good_numb_url" 1%}><i class="fa fa-envelope-o"
aria-hidden="true"></i></a></li>
{% endblock %}
```