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

Факультет «Системы обработки информации и управления» Кафедра ИУ5

Отчёт

## по лабораторной работе №4

по дисциплине «Разработка интернет приложений»

«Шаблонизация»

Студент: Акушко А.С. Группа РТ5-51

Преподаватель: Гапанюк Ю.Е.

Москва, 2019

**Задание и порядок выполнения ЛР №4**

1. В этой ЛР вы создадите Django-проект, покажете пользователю статичную страницу, познакомитесь с конструкциями шаблонизаторов: переменные, теги, наследование шаблонов.
2. Создать проект
3. Реализовать view, в которых генерируются html-страницы
4. В шаблонах должны быть использованы рассмотренные конструкции: переменные, вложенные значения, циклы, условия
5. Все шаблоны должны расширять базовый шаблон
6. Для элементов списка использовать тег include
7. По нажатии на элемент списка должна открываться страница информации об элементе
8. Для верстки необходимо использовать Bootstrap

**Код программы**

**urls.py**

*"""mylab URL Configuration*

*The `urlpatterns` list routes URLs to views. For more information please see: https://docs.djangoproject.com/en/3.0/topics/http/urls/*

*Examples:*

*Function views*

* 1. *Add an import: from my\_app import views*
  2. *Add a URL to urlpatterns: path('', views.home, name='home') Class-based views*

1. *Add an import: from other\_app.views import Home*
2. *Add a URL to urlpatterns: path('', Home.as\_view(), name='home') Including another URLconf*
3. *Import the include() function: from django.urls import include, path*
4. *Add a URL to urlpatterns: path('blog/', include('blog.urls'))*

*"""*

**from** django.contrib **import** admin **from** django.conf.urls **import** url **from** django.urls **import** path, include

**from** posts.views **import** PostView, ListPostView

urlpatterns = [

url(**r'^admin/'**, admin.site.urls),

url(**r'^post/(?P<post\_id>\d+)'**, PostView.as\_view()), *# (?P<post\_id>\d+)*

url(**r'^$'**, ListPostView.as\_view(), name=**'posts-list'**),

]

**base-template.html**

{% load static %}

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<!-

- The above 3 meta tags \*must\* come first in the head; any other head content must come \*after\* these ta gs -->

<title>{% block title %}{% endblock %}</title>

<!-- Bootstrap core CSS -->

<link href="{% static 'css/bootstrap.min.css' %}" rel="stylesheet">

<!-- Custom styles for this template -->

<style>

body {

padding-top: 60px; background-color: white;

}

.jumbotron {

margin: 20px -30px; padding: 20px 20px;

}

</style>

</head>

<body>

<div class="container">

<div class="starter-template">

{% block body %}

{% endblock %}

</div>

</div><!-- /.container -->

<!-- Bootstrap core JavaScript

================================================== -->

<!-- Placed at the end of the document so the pages load faster -->

<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.4/jquery.min.js"></script>

<script src="{% static 'js/bootstrap.min.js' %}"></script>

</body>

</html>

### post\_html.html

{% extends 'base\_template.html' %}

{% block title %} {% endblock %}

{% block navbar-links %}

<ul class="nav navbar-nav">

<li><a href="/"> back to list of all</a></li>

</ul>

{% endblock %}

{% block body %}

<h1>{{ post.post\_name }}</h1>

<div class="media">

<div class="media-left">

<a href="#">

<img class="media-

object" src="{{ post.post\_image }}" alt="{{ post.post\_name }}" height=250>

</a>

</div>

<div class="media-body">

<div class="jumbotron">

<h5>{{ post.pub\_data }}</h5>

<p>{{ post.post\_text }}</p>

</div>

<p>

&nbsp;

<a class="btn btn-lg btn-success" role="button">Read more</a>

</p>

</div>

</div>

{% endblock %}

### post\_list.html

{% extends 'base\_template.html' %}

{% block title %} {% endblock %}

{% block navbar-links %} {% endblock %}

{% block body %}

<h1>Posts</h1>

<ul>

{% for post in object\_list%}

{%include 'element.html' with post=post %}

{% endfor %}

</ul>

{% endblock %}

### element.html

<div class="jumbotron">

<div class="media-left media-middle">

<a href="#">

<img class="media-

object" src="{{ post.post\_image }}" alt="{{ post.post\_name }}" width=200>

</a>

</div>

<div class="media-body">

<br/>

<h2>{{ post.post\_name }}</h2>

<p><span class="label label-default">{{ post.pub\_data }}</span></p>

<p>{{ post.post\_text }}</p>

<p><a class="btn btn-primary btn-

lg" href="post/{{ post.id }}" role="button">Learn more</a></p>

</div>

</div>

### views.py

**from** django.http **import** HttpResponse

**from** django.shortcuts **import** render

**from** django.views.generic **import** View, ListView

**from** posts.models **import** Post

**from** django.http **import** HttpResponse

**def** func\_view(request):

**return** HttpResponse(**"Function view response"**)

**class** ListPostView(ListView): model = Post

template\_name = **'post\_list.html'**

**class** PostView(View):

**def** get(self, request, post\_id):

post = Post.objects.get(id=post\_id)

**return** render(request, **'post\_html.html'**, {**"post"**: post})

### models.py

**from** django.db **import** models

**class** Post(models.Model):

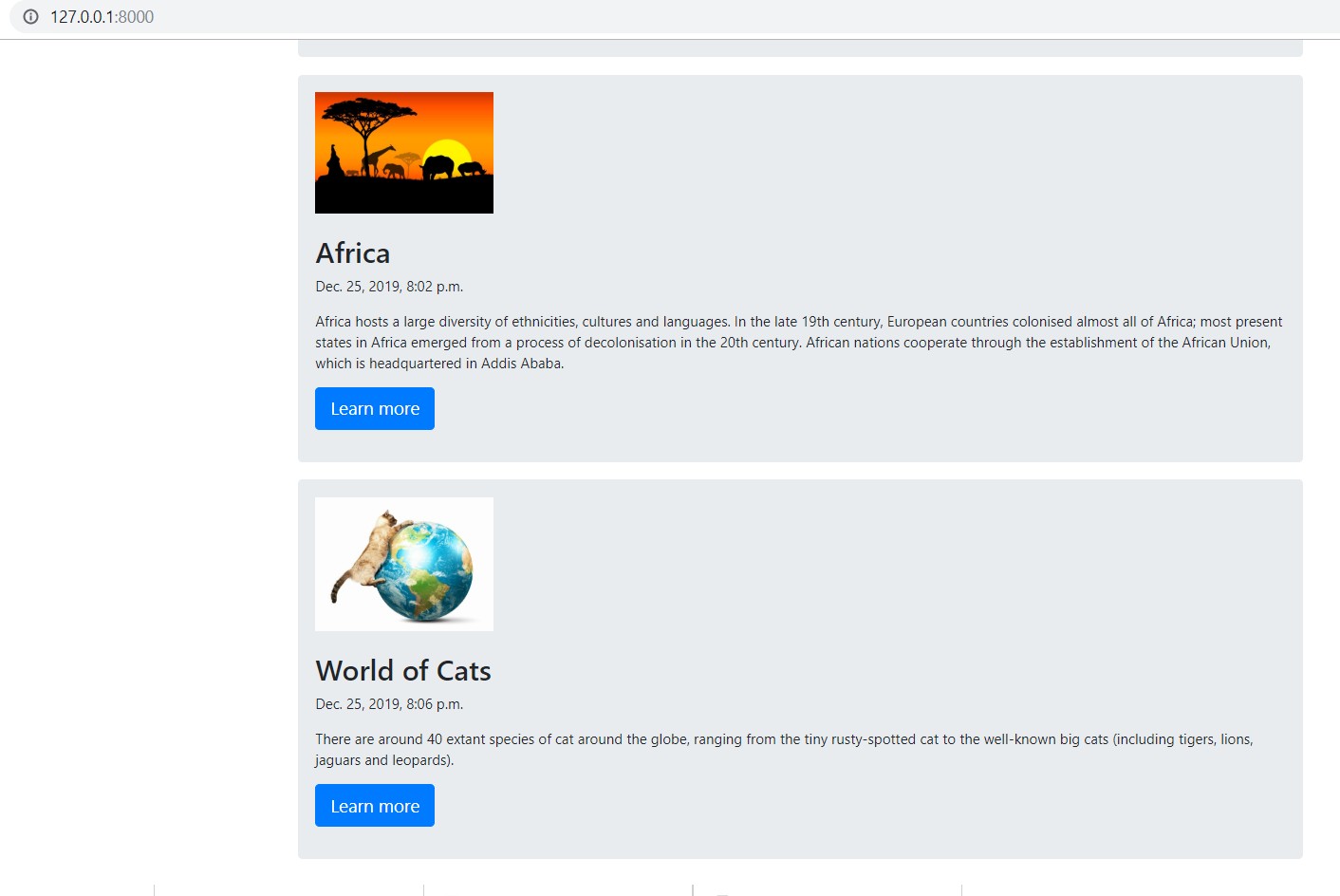
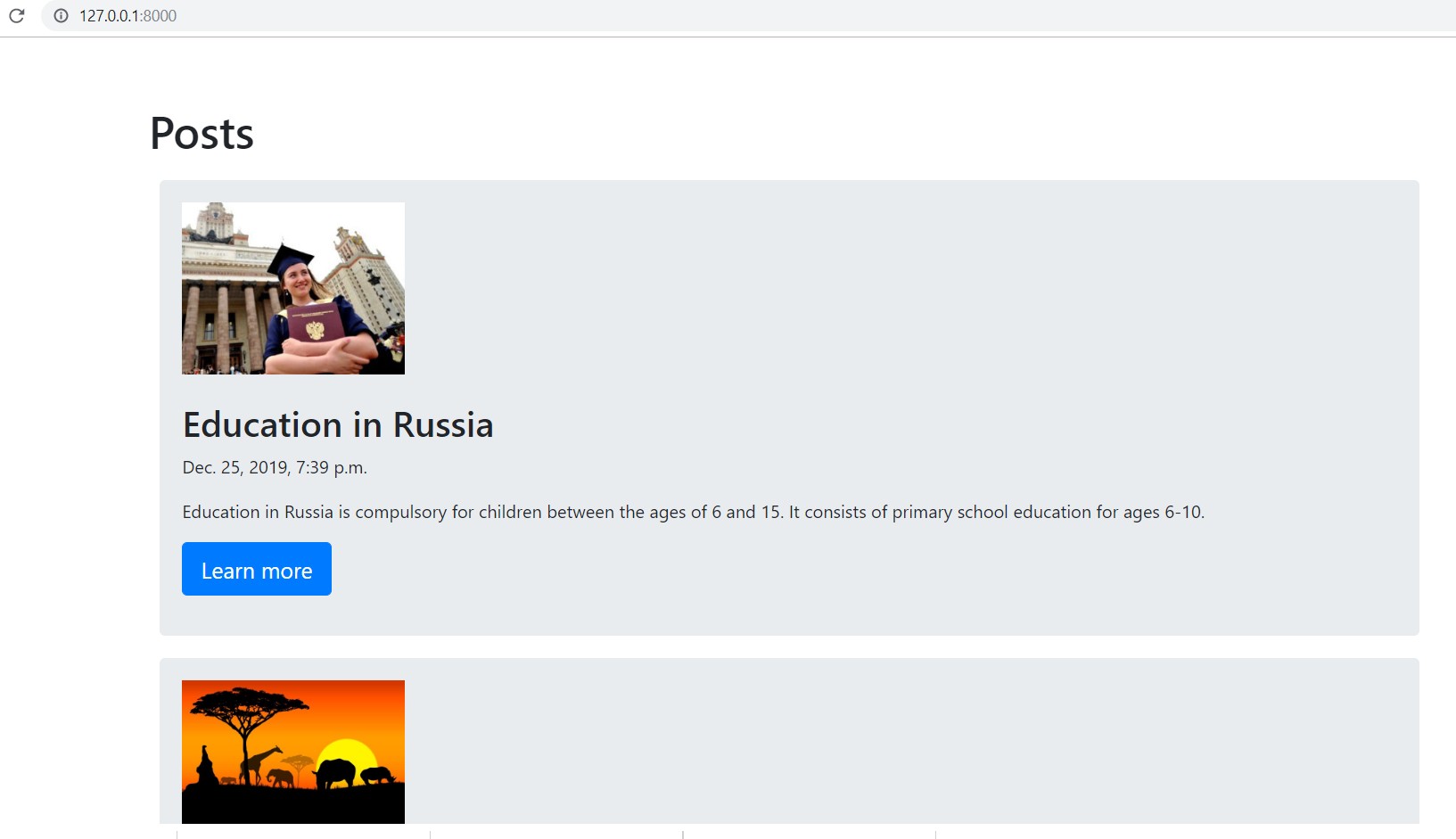
post\_name = models.CharField(**'Заголовок поста'**, max\_length=255) post\_text = models.CharField(**'Текст поста'**, max\_length=500) pub\_data = models.DateTimeField(**'Дата публикации'**)

post\_image = models.URLField(**'Изображение'**, max\_length=255)

**def** str (self):

**return 'Post {}'**.format(self.post\_name)

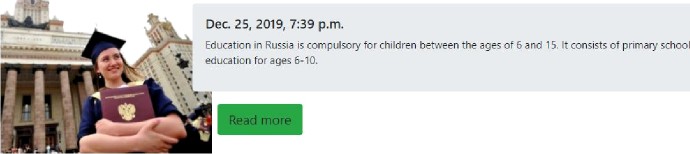
# Скриншоты выполнения



@ 127.0.0.1:806 /post/1

{g 127.0.0.1:806/post/2

## Education in Russia



Africa

Dec. 25, 2019, 8:02 p.m.

Africa hosls а large diversity of ethnicities cuftures and lanдuages. In the late 19th century, Europeaп auntries colonised almost all of AFrica; most presenE states in Africa emerдed from а pюcess oF decolonisation'in the 20th century. African nations cooperate through the estaЫishment of the African Union, which s headquartered Tn Addis Ababa.

Ы 127.0.0.1:8000/post/3

## World of Cats

Dec. 25. 2019, 8:06 p.m.

There are.around 40 extant species Of cat around the globe, ranging fmom the tiny rusty-spotted cat to the well- kлawn big cats (including tigers, lions, yaguars and leopards).





[2f..'’Eec..'’2G19 G2:21:36] ”бБТ '’ HTTP..'’1.1“ 2GG 373Б

C:’..Use’s’..belun’..Eccu\*en's’..PliП’...ab\_$’.. ylab’..\*ylab’..apps’..pcs\*s’..ad in.p,' changed, ’elc ading.

›.'a'ching Tc’ lile changes ...'i'h S'a'Relcade- Pe’Ic’ring syste checks...

S з\*ег check і den \* і f і е d nc ia auea it С аі 1 enc е d ) . С е сег b е 2E , 2C19 C2 : 2 9 : 15

Cj angc 've “зі on 3 . С . 1, из ing з e"\*ingз ’ п 1аЬ . зе "" 1 ng з ’

S " а " і ng de'v е1 срп en \* з е 'v е а " h "" р : ’'..’'1 2 7 . С . С . 1 : 6 С С С..’'

¿uï" "hе зе 've т'.'i•h CTR - Вя EïK .

[ 2Е..’' С е с ..’' 2С1 9 С 2 : Э С : 27 ] " G Е Т ..’' Д с з "..’” 2 HTTP..’' 1 . 1 " 2С С 21 26

[ 2E..' Cec..' 2C19 C2 : ЗС : 5C ] " G ЕТ ..' рев+..'”1 HTTP..' 1 . 1 " 2СС 1951

[ 2E..’ Cec..’ 2C19 C2 ! 31 ! 13 ] " G "Т ’ рсз\*..’”3 HTTP..’ 1 . 1 " 2СС 1956

С : ‘',Use a‘',be1un‘',Cccur en\*a ‘',P £1П‘', .ab\_4‘',г 1ab‘',г 1аЬ‘',u 1s . pt' сhange d , е1с ading . r.’atching le life change а i.ith 5 t а tRe1с ade

Р е f с п і ng з з\* еп с hе сk з ...

Т d 2

sues 'jG silenced).

Е. e’':Бk 2d1 "

„ '

а ng’de've с ’ nt’se ve’ а\*’ #р''’!l27.G: G.1' GG

Qui" "he se’ve’ ’.d"h CTR.-BRYK.

С: \Use’s\belun\Eccuз en's\PliП\.аЬ $’,v,lab›