**Московский государственный технический**

**университет им. Н.Э. Баумана.**

Факультет «Информатика и управление»

Кафедра ИУ5.

Курс «Разработка интернет-приложений»

Отчет по лабораторной работе №4-5

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студент группы ИУ5-52 |  | преподаватель каф. ИУ5 |
| Король Константин |  | Гапанюк Ю.Е. |
| Подпись и дата: |  | Подпись и дата: |

Москва, 2018 г.

**Задание.**

**Задание.**

* 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

**Код программы**

Mysite/urls.py

from django.contrib import admin  
from django.urls import include, path  
  
urlpatterns = [  
 path('admin/', admin.site.urls),  
 path('', include('polls.urls')),  
  
  
]

polls/admin.py

from django.contrib import admin  
from .models import Country, News  
  
admin.site.register(Country)  
admin.site.register(News)

polls/models.py

from django.db import models  
  
  
class Country(models.Model):  
 country\_name = models.CharField(max\_length=200)  
 def \_\_str\_\_(self):  
 return self.country\_name  
  
  
class News(models.Model):  
 country = models.ForeignKey(Country, on\_delete=models.CASCADE)  
 news\_text = models.CharField(max\_length=200)  
 pub\_date = models.DateTimeField('date published')  
 def \_\_str\_\_(self):  
 return self.news\_text

polls/urls.py

from django.urls import path  
  
from . import views  
  
urlpatterns = [  
 path('', views.chosed\_contry, name='chosed\_contry'),  
 #path('<int:country\_name>/chosed\_contry/', views.chosed\_contry, name='chosed\_contry'),  
 path('index.html', views.index, name='index'),  
 path('child/', views.Lala.as\_view()),  
]

polls/views.py

from django.views import View  
  
  
def chosed\_contry(request):  
 country\_list = Country.objects.order\_by('country\_name')  
 template = loader.get\_template('polls/chosed\_contry.html')  
 context = {  
 'country\_list':country\_list,  
 }  
 return HttpResponse(template.render(context, request))  
  
  
def index(request):  
 latest\_news\_list = News.objects.order\_by('pub\_date')[:5]  
 template = loader.get\_template('polls/index.html')  
 context = {  
 'latest\_news\_list':latest\_news\_list,  
 }  
 return HttpResponse(template.render(context, request))  
  
class Lala(View):  
 def get(self, request):  
 return render(request, 'Dukechild.html')

**Шаблоны**

Chosed\_contry.html

<!DOCTYPE html>  
<html lang="en">  
<head>  
 <meta charset="UTF-8">  
 <title>World News</title>  
 <link rel="stylesheet" href="//maxcdn.bootstrapcdn.com/bootstrap/3.2.0/css/bootstrap.min.css">  
 <link rel="stylesheet" href="//maxcdn.bootstrapcdn.com/bootstrap/3.2.0/css/bootstrap-theme.min.css">  
</head>  
<body>  
 <div>  
 {%block body%}  
 {% if country\_list%}  
 <ul>  
 {% for country in country\_list %}  
 <li><a href="index.html">{{country.country\_name}}</a></li>  
 {% endfor %}  
 </ul>  
 {% else %}  
 <p>No polls are available</p>  
 {% endif %}  
 {%endblock%}  
 </div>  
</body>  
</html>

Index.html

<!DOCTYPE html>  
<html lang="en">  
<head>  
 <meta charset="UTF-8">  
 <title>News</title>  
</head>  
<body>  
 {% if latest\_news\_list%}  
 <ul>  
 {% for news in latest\_news\_list %}  
 <li><a>{{news.news\_text}}</a></li>  
 {% endfor %}  
 </ul>  
 {% else %}  
 <p>No polls are available</p>  
 {% endif %}  
</body>  
</html>

Duke.html

<!DOCTYPE html>  
<html lang="en">  
<head>  
 <meta charset="UTF-8">  
 <title>Title</title>  
</head>  
<body>  
 <div>  
 {%block body%}  
 Это родитель  
 {% endblock body%}  
 </div>  
</body>  
</html>

Dukechild.html

{%extends "Duke.html"%}  
 <div>  
 {%block body%}  
 Это наследник  
 {%endblock body%}  
 </div>

Результаты





