**Отчёт по лабораторной работе №4 по курсу «Разработка интернет приложений»**

Тема: «Python. Шаблонизация»

Выполнил: студент группы ИУ5-53

Андреев Алексей Андреевич

Дата: 16/10/2018 Подпись:\_\_\_\_\_\_\_

Проверил: Гапанюк Ю. Е.

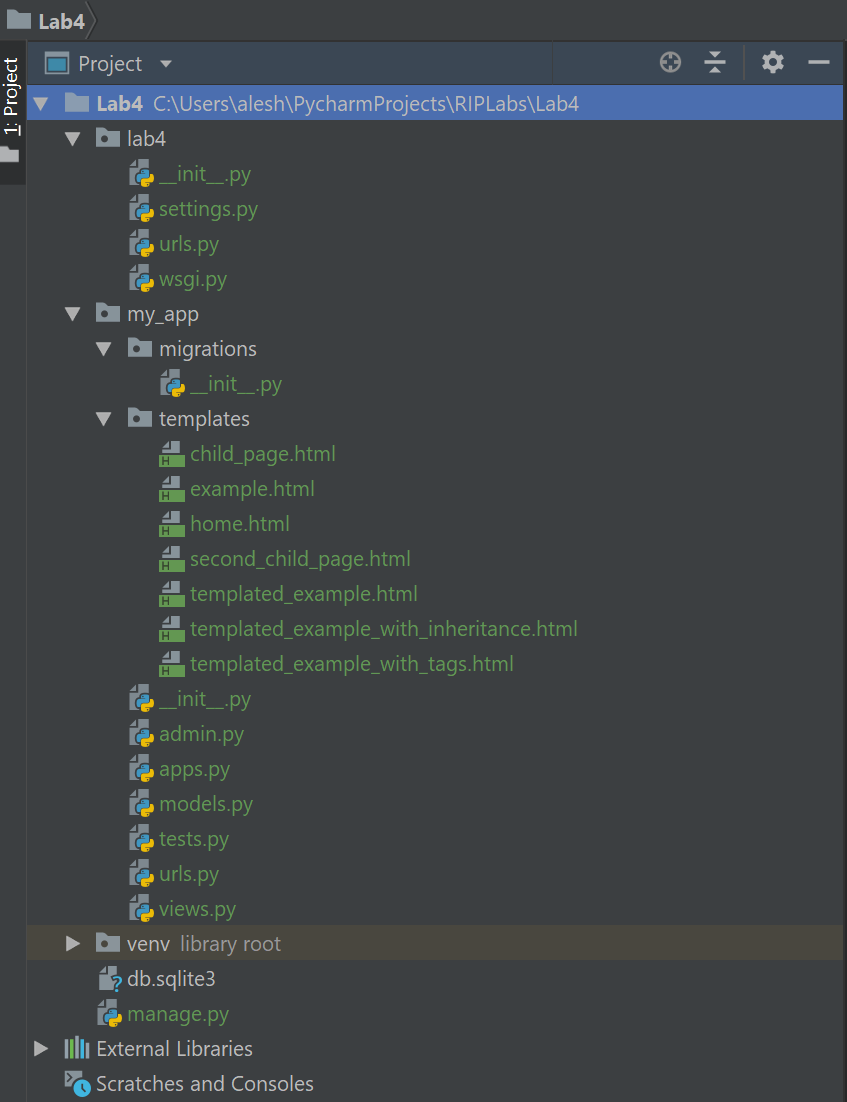
Дата: \_\_\_\_\_\_ Подпись:\_\_\_\_\_\_\_

Москва, 2018

Задание и порядок выполнения

* 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



Исходный код

Lab4/urls.py

*"""lab4 URL Configuration  
  
The `urlpatterns` list routes URLs to views. For more information please see:  
 https://docs.djangoproject.com/en/2.1/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  
 1. Import the include() function: from django.urls import include, path  
 2. Add a URL to urlpatterns: path('blog/', include('blog.urls'))  
"""*from django.contrib import admin  
from django.urls import include, path  
  
urlpatterns = [  
 path('', include('my\_app.urls')),  
 path('admin/', admin.site.urls),  
]

my\_app/urls.py

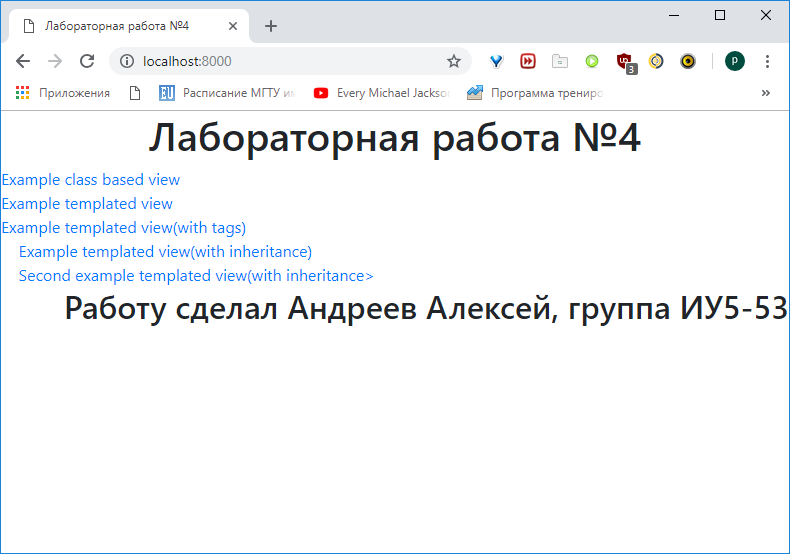
from django.urls import path  
  
from . import views  
  
  
urlpatterns = [  
 path('', views.index),  
 path('example\_class\_based/', views.ExampleClassBased.as\_view()),  
 path('example\_static/', views.ExampleStaticClassBased.as\_view()),  
 path('example\_templated\_view/', views.ExampleTemplatedView.as\_view()),  
 path('templated\_example\_with\_tags/', views.ExampleTemplatedViewWithTags.as\_view()),  
 path('templated\_example\_with\_inheritance/', views.ExampleTempledViewWithInheritance.as\_view()),  
 path('second\_templated\_example\_with\_inheritance/', views.ExampleTempledSecondViewWithInheritance.as\_view()),  
]

views.py

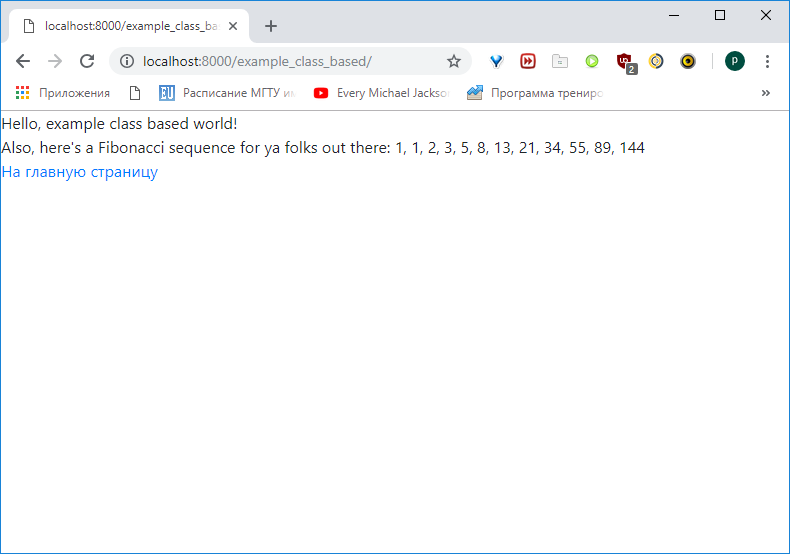
from django.shortcuts import render  
from django.http import HttpResponse  
from django.views.generic import View  
# Create your views here.  
  
from random import randint  
  
def index(request):  
 return render(request, 'home.html')  
  
  
def CountFibonacci():  
 first\_number = 1  
 second\_number = 1  
 return\_list = [first\_number, second\_number]  
  
 for i in range(10):  
 third\_number = first\_number + second\_number  
 first\_number = second\_number  
 second\_number = third\_number  
 return\_list.append(third\_number)  
 return str(return\_list)[1:-1]  
  
  
def CountFibonacciList():  
 first\_number = 1  
 second\_number = 1  
 return\_list = [first\_number, second\_number]  
  
 for i in range(10):  
 third\_number = first\_number + second\_number  
 first\_number = second\_number  
 second\_number = third\_number  
 return\_list.append(third\_number)  
 return return\_list  
  
  
class ExampleClassBased(View):  
 def get(self, request):  
 return HttpResponse(f"""  
 <head>  
 <meta name="viewport" content="width=device-width, initial-scale=1">  
 <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css">  
 </head>  
 Hello, example class based world!  
 <br>  
 Also, here's a Fibonacci sequence for ya folks out there: {CountFibonacci()}  
 <br>  
 <a href="/">На главную страницу</a>  
 """)  
  
  
class ExampleStaticClassBased(View):  
 def get(self, request):  
 return render(request, 'example.html')  
  
  
class ExampleTemplatedView(View):  
 def get(self, request):  
 return render(request, 'templated\_example.html', { 'some\_variable' : str(randint(0, 100))})  
  
  
class ExampleTemplatedViewWithTags(View):  
 def get(self, request):  
 return render(request, 'templated\_example\_with\_tags.html', {'list' : CountFibonacciList()})  
  
  
class ExampleTempledViewWithInheritance(View):  
 def get(self, request):  
 return render(request, 'child\_page.html', {'elements' : [randint(1,100) for i in range(10)]})  
  
  
  
class ExampleTempledSecondViewWithInheritance(View):  
 def get(self, request):  
 return render(request, 'second\_child\_page.html', {'elements' : [i for i in range(10)]})

Результат работы

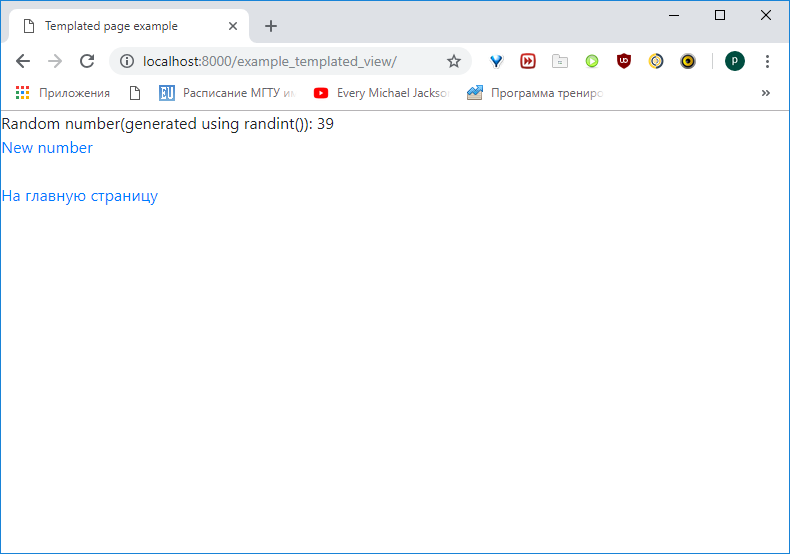
Домашняя страница



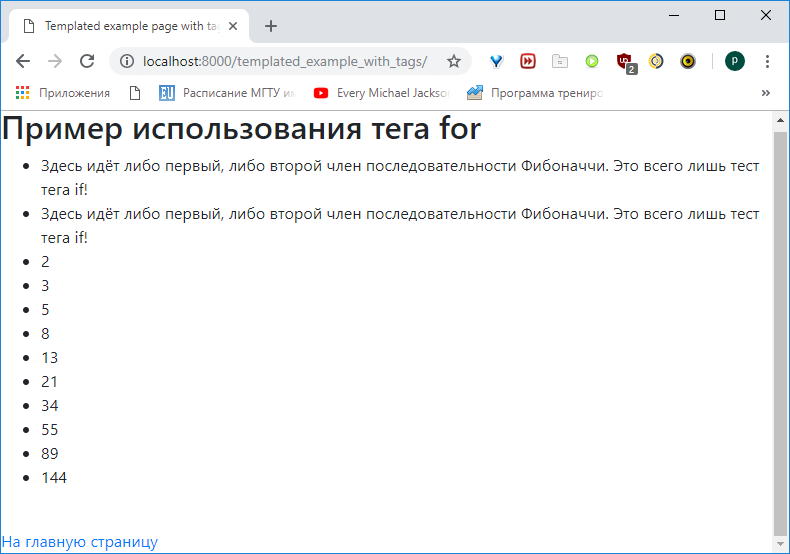
Пример class-based view



Пример шаблонного view с одной переменной



Пример шаблонного view с тегами



Пример шаблонных view с наследованием

