Отчет по лабораторной работе № 5 по курсу

Разработка интернет-приложений

Тема работы: "Шаблонизация"

5

(количество листов)

Выполнила:

студентка группы ИУ5-53

Костенкова Юлия

Москва, МГТУ - 2017

СОДЕРЖАНИЕ

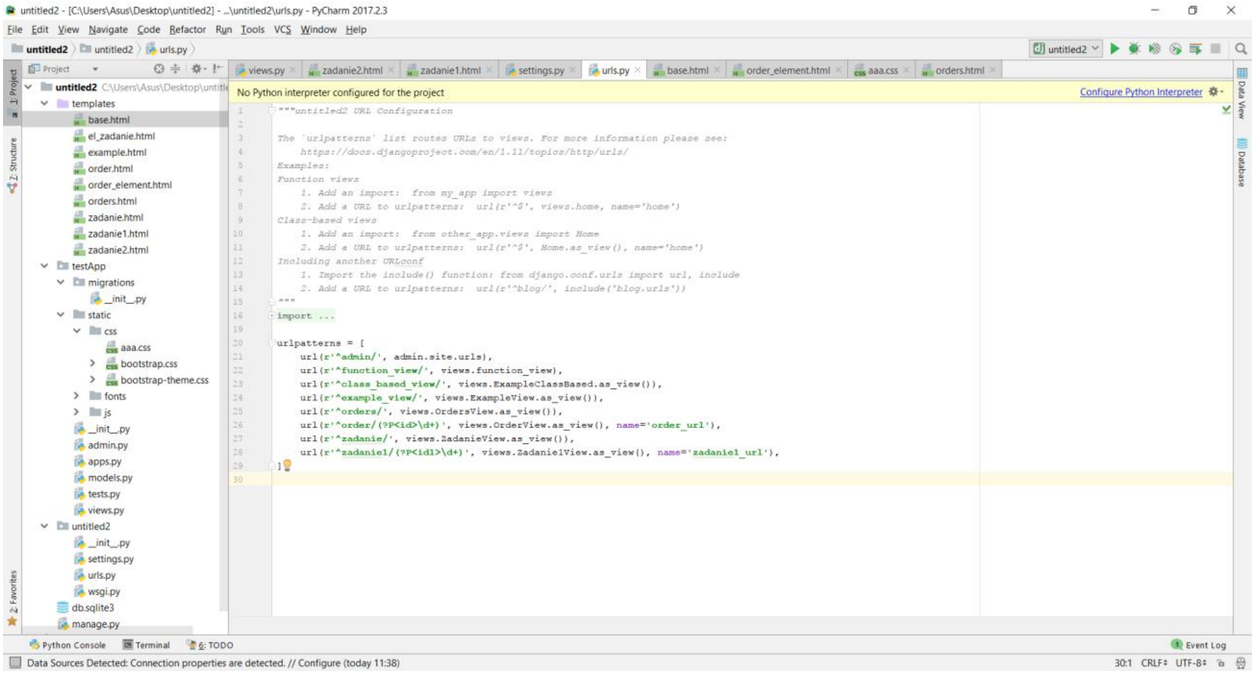
Задание и порядок выполнения

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

Код программы:

Urls.py

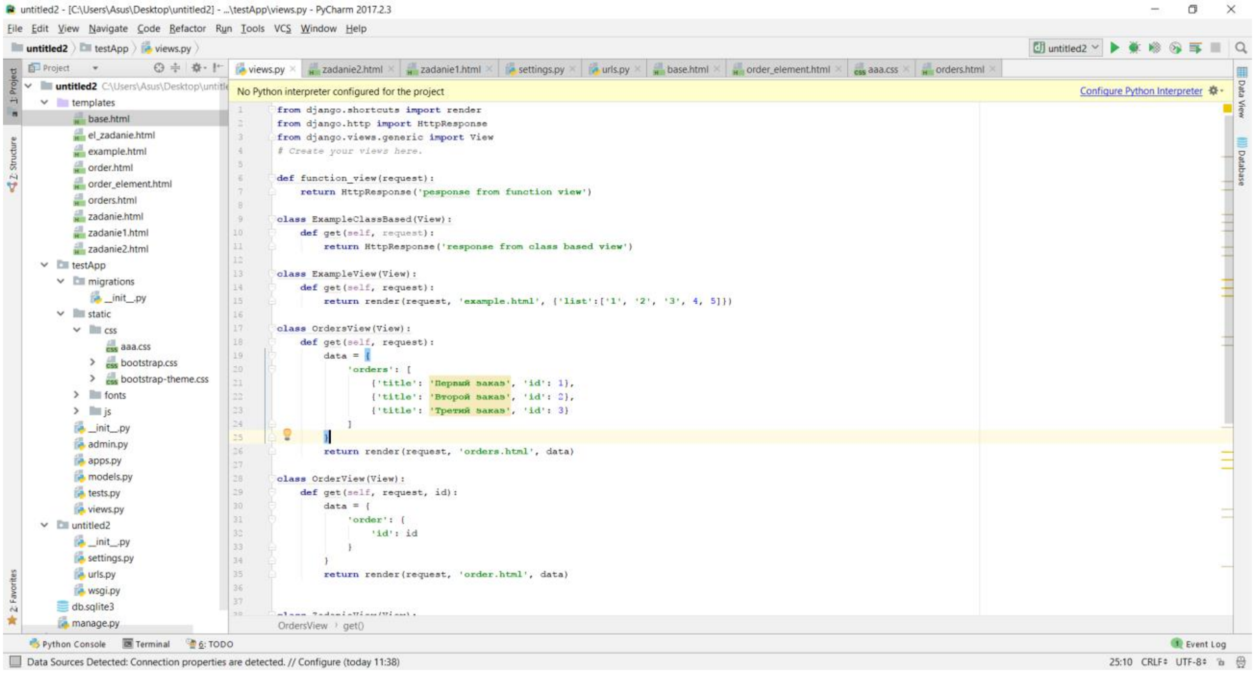


No Python interpreter configured for the project

```python
"""untitled2 URL Configuration

The `urlpatterns` list routes URLs to views. For more information please see:
    https://docs.djangoproject.com/en/1.11/topics/http/urls/
Examples:
Function views
    1. Add an import:  from my_app import views
    2. Add a URL to urlpatterns:  url(r'^$', views.home, name='home')
Class-based views
    1. Add an import:  from other_app.views import Home
    2. Add a URL to urlpatterns:  url(r'^$', Home.as_view(), name='home')
Including another URLconf
    1. Import the include() function: from django.conf.urls import url, include
    2. Add a URL to urlpatterns:  url(r'^blog/', include('blog.urls'))
"""
import ...

urlpatterns = [
    url(r'^admin/', admin.site.urls),
    url(r'^function_view/', views.function_view),
    url(r'^class_based_view/', views.ExampleClassBased.as_view()),
    url(r'^example_view/', views.ExampleView.as_view()),
    url(r'^orders/', views.OrdersView.as_view()),
    url(r'^order/(?P<id>\d+)', views.OrderView.as_view(), name='order_url'),
    url(r'^zadanie/', views.ZadanieView.as_view()),
    url(r'^zadanie1/(?P<id1>\d+)', views.Zadanie1View.as_view(), name='zadanie1_url'),
]
```

Data Sources Detected: Connection properties are detected. // Configure (today 11:38)

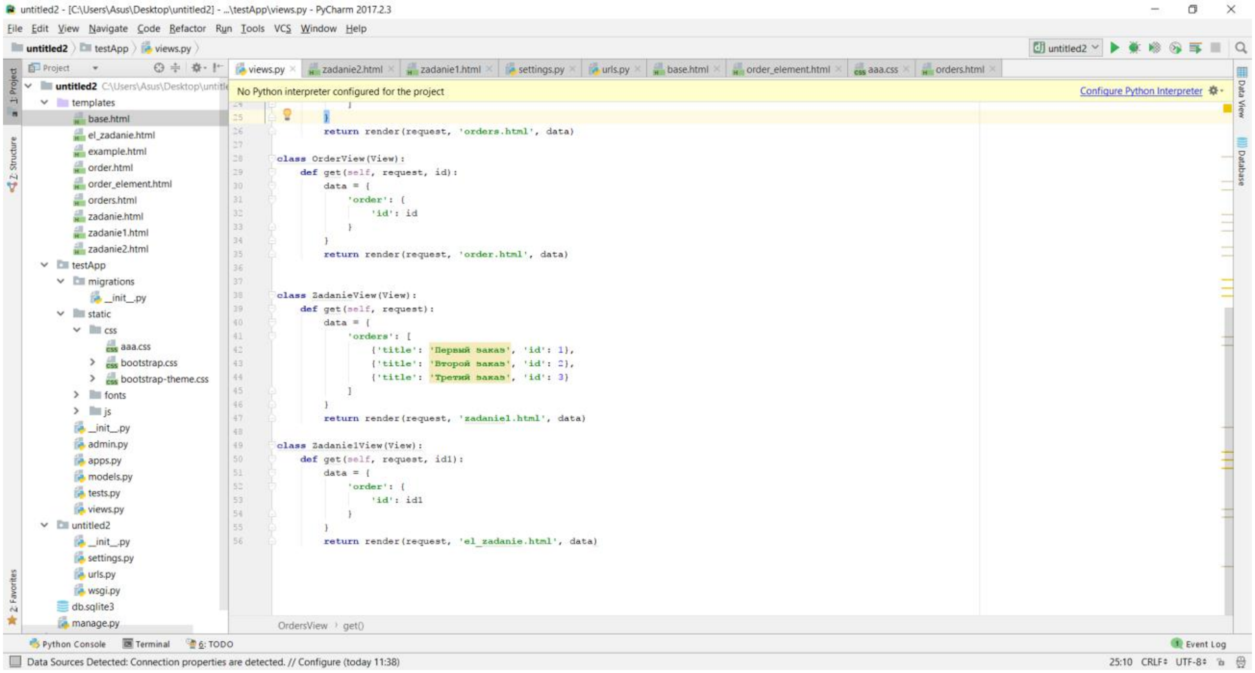
Views.py


untitled2 - [C:\Users\Asus\Desktop\untitled2] - ...\testApp\views.py - PyCharm 2017.2.3
File Edit View Navigate Code Refactor Run Tools VCS Window Help
untitled2 〉 testApp 〉 views.py
Project
views.py
zadanie2.html
zadanie1.html
settings.py
urls.py
base.html
order_element.html
aaa.css
orders.html
Configure Python Interpreter
OrdersView 〉 get()
Python Console
Terminal
6: TODO
Event Log
25:10 CRLF UTF-8

No Python interpreter configured for the project

```python
from django.shortcuts import render
from django.http import HttpResponse
from django.views.generic import View
# Create your views here.

def function_view(request):
    return HttpResponse('pesponse from function view')

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

class ExampleView(View):
    def get(self, request):
        return render(request, 'example.html', {'list':['1', '2', '3', 4, 5]})

class OrdersView(View):
    def get(self, request):
        data = {
            'orders': [
                {'title': 'Первый заказ', 'id': 1},
                {'title': 'Второй заказ', 'id': 2},
                {'title': 'Третий заказ', 'id': 3}
            ]
        }
        return render(request, 'orders.html', data)

class OrderView(View):
    def get(self, request, id):
        data = {
            'order': {
                'id': id
            }
        }
        return render(request, 'order.html', data)


class ZadanieView(View):
```

Data Sources Detected: Connection properties are detected. // Configure (today 11:38)



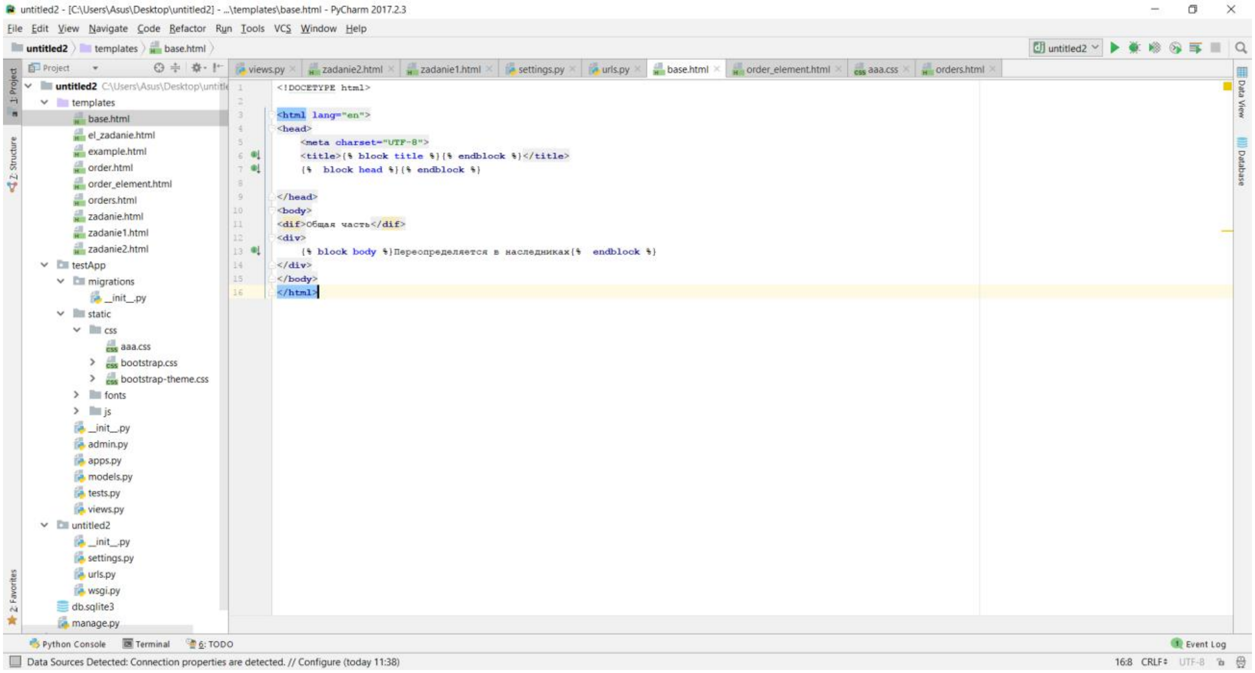
untitled2 - [C:\Users\Asus\Desktop\untitled2] - ...\testApp\views.py - PyCharm 2017.2.3

No Python interpreter configured for the project

```
            ]
        }
        return render(request, 'orders.html', data)

class OrderView(View):
    def get(self, request, id):
        data = {
            'order': {
                'id': id
            }
        }
        return render(request, 'order.html', data)


class ZadanieView(View):
    def get(self, request):
        data = {
            'orders': [
                {'title': 'Первый заказ', 'id': 1},
                {'title': 'Второй заказ', 'id': 2},
                {'title': 'Третий заказ', 'id': 3}
            ]
        }
        return render(request, 'zadanie1.html', data)

class Zadanie1View(View):
    def get(self, request, id1):
        data = {
            'order': {
                'id': id1
            }
        }
        return render(request, 'el_zadanie.html', data)
```

Data Sources Detected: Connection properties are detected. // Configure (today 11:38)

Base.html

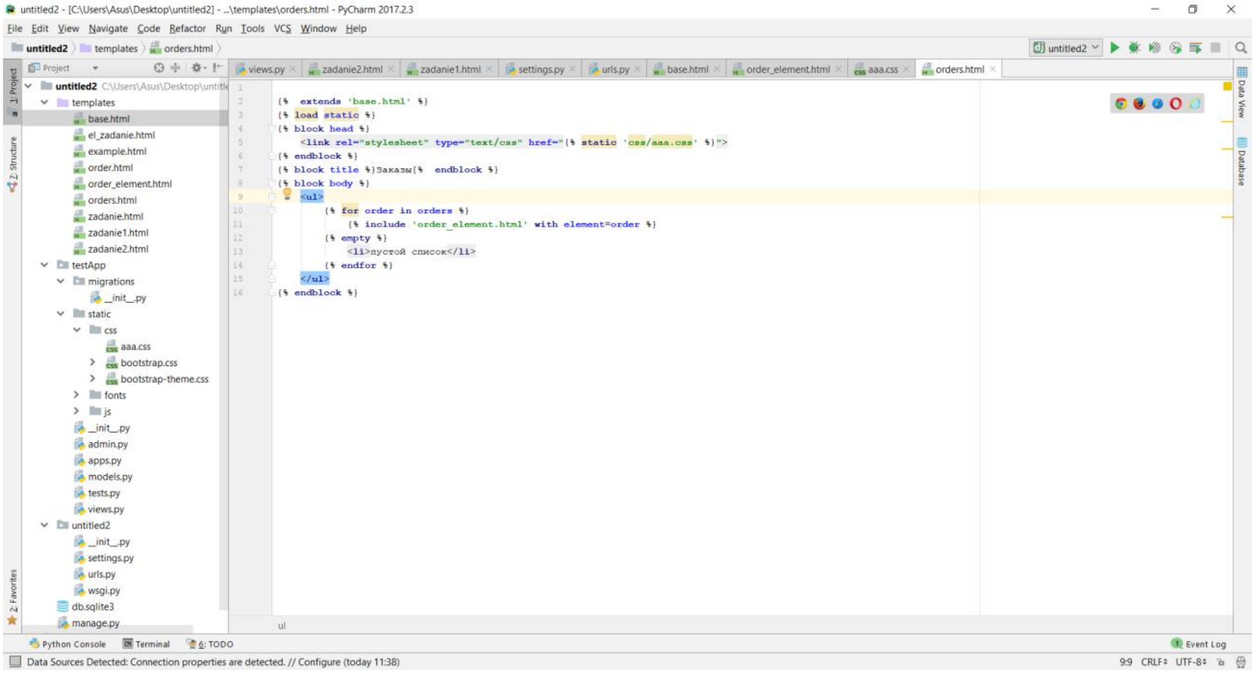


untitled2 - [C:\Users\Asus\Desktop\untitled2] - ...\templates\base.html - PyCharm 2017.2.3

```
<!DOCETYPE html>

<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>{% block title %}{% endblock %}</title>
    {%  block head %}{% endblock %}

</head>
<body>
<dif>Общая часть</dif>
<div>
    {% block body %}Переопределяется в наследниках{%  endblock %}
</div>
</body>
</html>
```

Data Sources Detected: Connection properties are detected. // Configure (today 11:38)

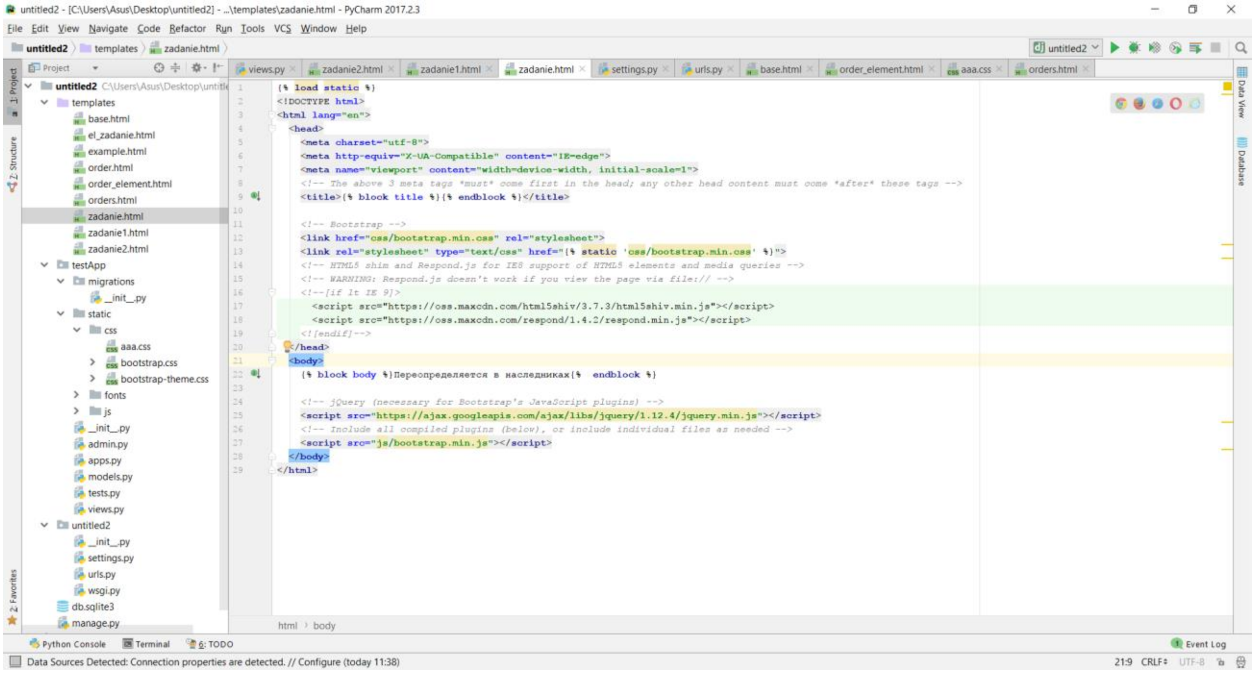
Orders.html



```
{%  extends 'base.html' %}
{% load static %}
{% block head %}
    <link rel="stylesheet" type="text/css" href="{% static 'css/aaa.css' %}">
{% endblock %}
{% block title %}Заказы{%  endblock %}
{% block body %}
    <ul>
        {% for order in orders %}
            {% include 'order_element.html' with element=order %}
        {% empty %}
            <li>пустой список</li>
        {% endfor %}
    </ul>
{% endblock %}
```

Data Sources Detected: Connection properties are detected. // Configure (today 11:38)

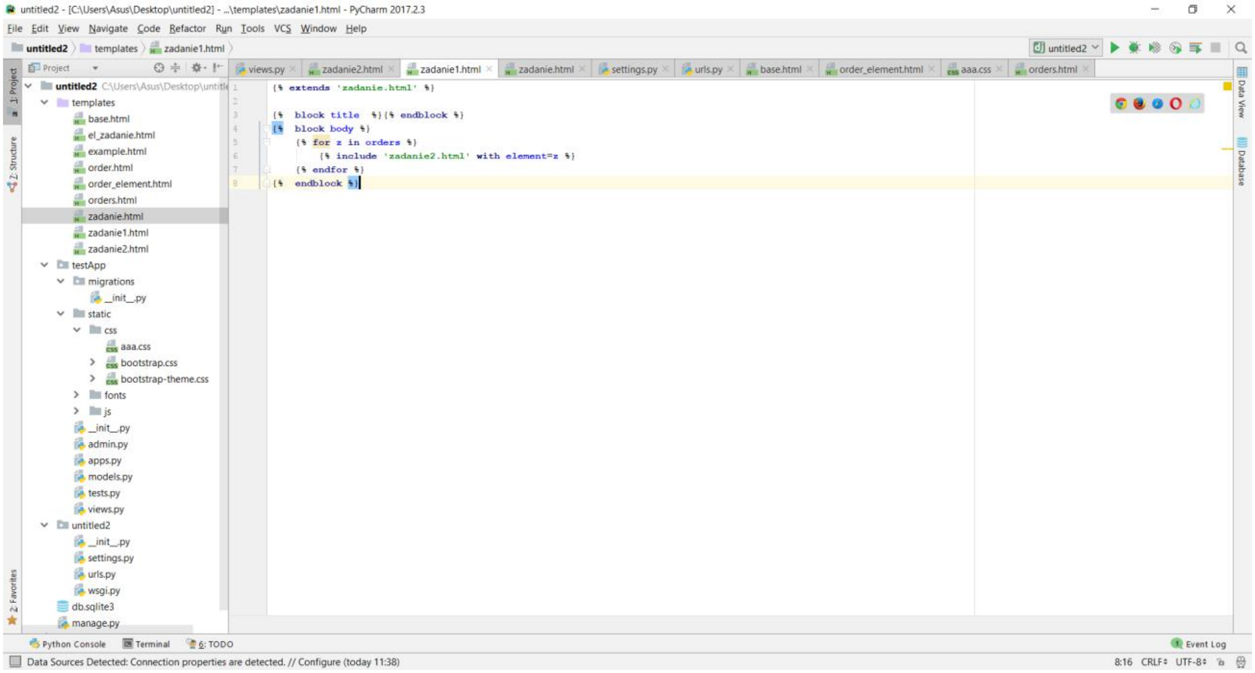
Zadanie.html


untitled2 - [C:\Users\Asus\Desktop\untitled2] - ...\templates\zadanie.html - PyCharm 2017.2.3
File Edit View Navigate Code Refactor Run Tools VCS Window Help
untitled2 〉 templates 〉 zadanie.html
views.py
zadanie2.html
zadanie1.html
zadanie.html
settings.py
urls.py
base.html
order_element.html
aaa.css
orders.html
html 〉 body
Python Console
Terminal
6: TODO
Event Log
21:9 CRLF UTF-8

```
{% 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 tags -->
    <title>{% block title %}{% endblock %}</title>

    <!-- Bootstrap -->
    <link href="css/bootstrap.min.css" rel="stylesheet">
    <link rel="stylesheet" type="text/css" href="{% static 'css/bootstrap.min.css' %}">
    <!-- HTML5 shim and Respond.js for IE8 support of HTML5 elements and media queries -->
    <!-- WARNING: Respond.js doesn't work if you view the page via file:// -->
    <!--[if lt IE 9]>
      <script src="https://oss.maxcdn.com/html5shiv/3.7.3/html5shiv.min.js"></script>
      <script src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>
    <![endif]-->
  </head>
  <body>
    {% block body %}Переопределяется в наследниках{%  endblock %}

    <!-- jQuery (necessary for Bootstrap's JavaScript plugins) -->
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.4/jquery.min.js"></script>
    <!-- Include all compiled plugins (below), or include individual files as needed -->
    <script src="js/bootstrap.min.js"></script>
  </body>
</html>
```

Data Sources Detected: Connection properties are detected. // Configure (today 11:38)

Zadanie1.html

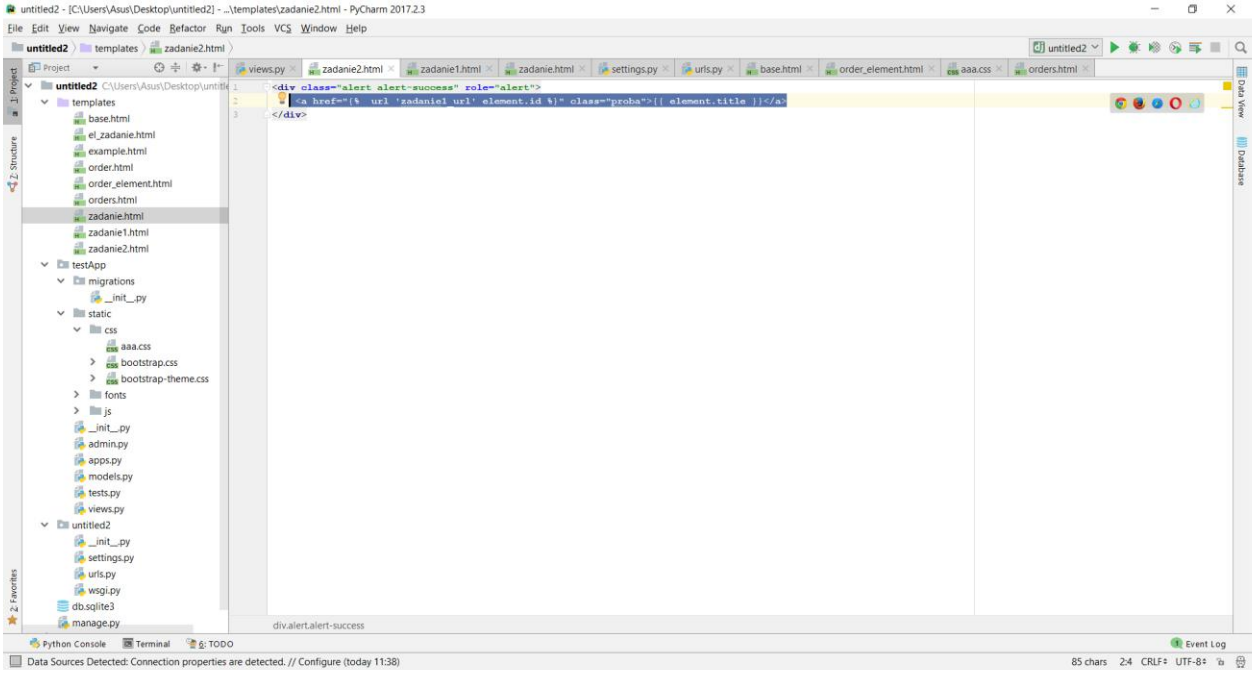


untitled2 - [C:\Users\Asus\Desktop\untitled2] - ...\templates\zadanie1.html - PyCharm 2017.2.3

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

{%  block title  %}{% endblock %}
{%  block body %}
    {% for z in orders %}
        {% include 'zadanie2.html' with element=z %}
    {% endfor %}
{%  endblock %}
```

Data Sources Detected: Connection properties are detected. // Configure (today 11:38)

Zadanie2.html

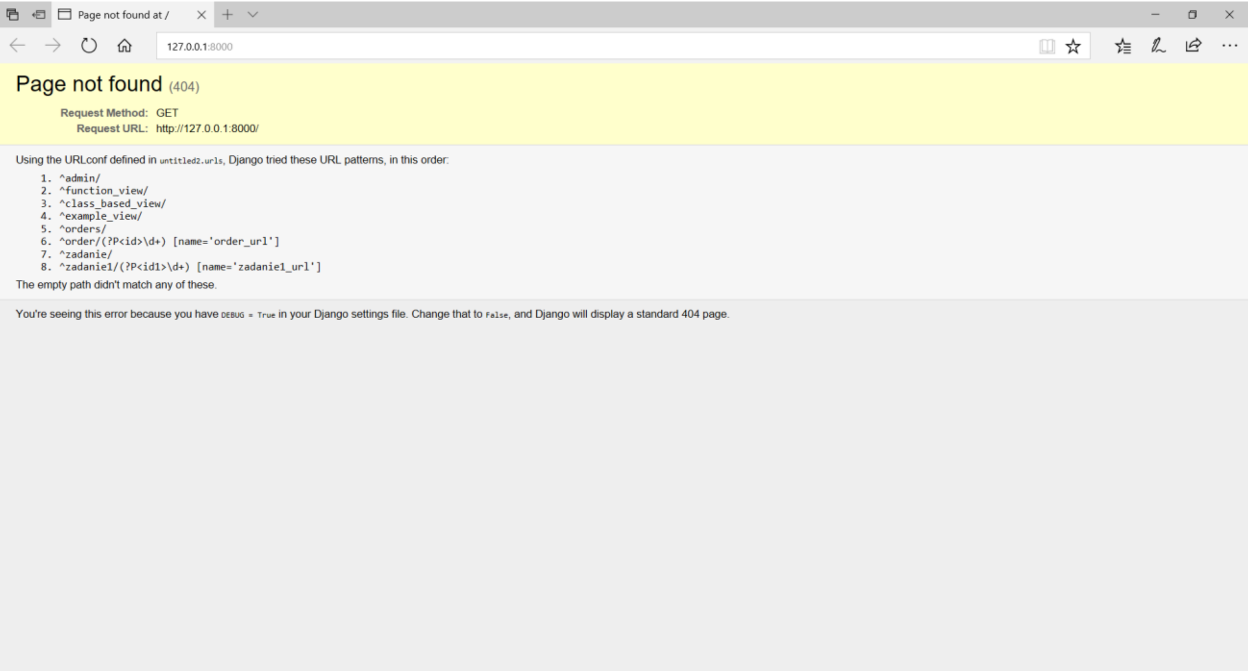


untitled2 - [C:\Users\Asus\Desktop\untitled2] - ...\templates\zadanie2.html - PyCharm 2017.2.3

```
<div class="alert alert-success" role="alert">
    <a href="{%  url 'zadanie1_url' element.id %}" class="proba">{{ element.title }}</a>
</div>
```

Data Sources Detected: Connection properties are detected. // Configure (today 11:38)

Результаты:



Page not found (404)

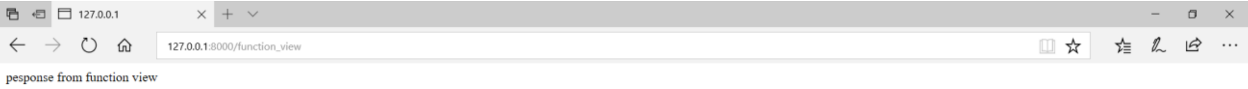
```
Request Method:  GET
   Request URL:  http://127.0.0.1:8000/
```

Using the URLconf defined in `untitled2.urls`, Django tried these URL patterns, in this order:

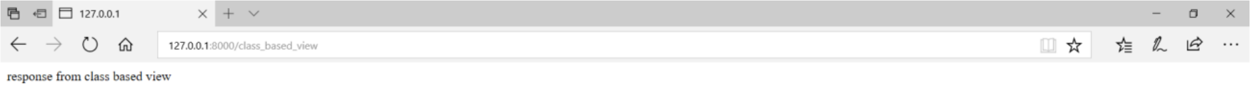
```
1. ^admin/
2. ^function_view/
3. ^class_based_view/
4. ^example_view/
5. ^orders/
6. ^order/(?P<id>\d+) [name='order_url']
7. ^zadanie/
8. ^zadanie1/(?P<id1>\d+) [name='zadanie1_url']
```

The empty path didn't match any of these.

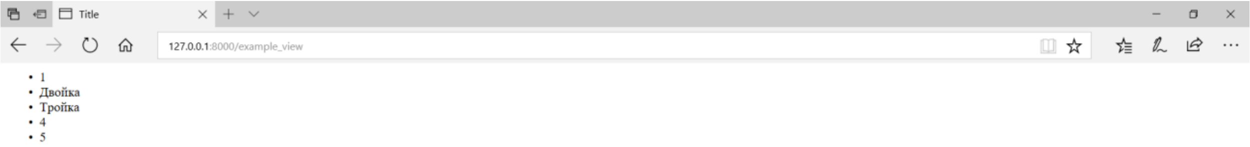
You're seeing this error because you have `DEBUG = True` in your Django settings file. Change that to `False`, and Django will display a standard 404 page.



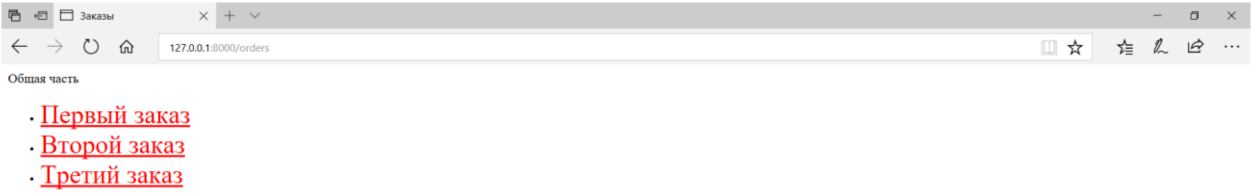
pesponse from function view



response from class based view

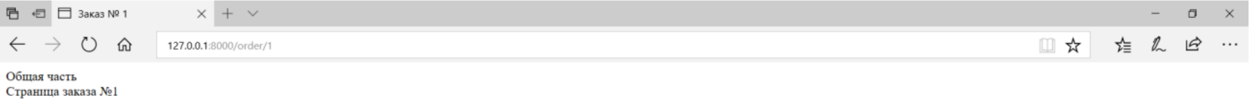


```
• 1
• Двойка
• Тройка
• 4
• 5
```

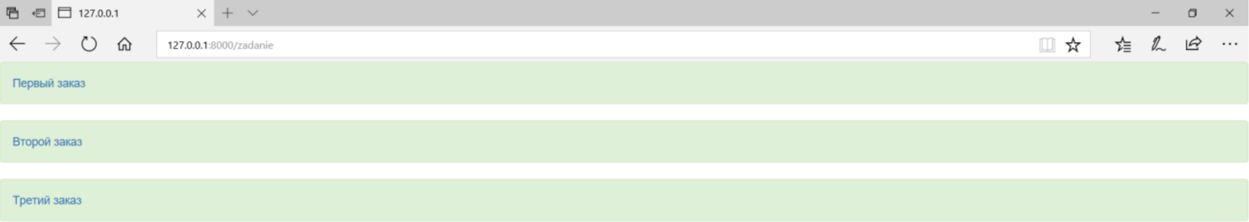


Общая часть

- Первый заказ
- Второй заказ
- Третий заказ



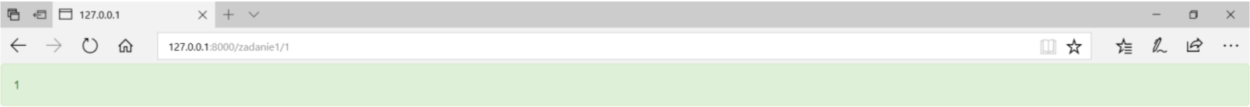
Общая часть
Страница заказа №1



Первый заказ

Второй заказ

Третий заказ



```
1
```