МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное

образовательное учреждение высшего образования

«НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет АВТОМАТИКИ И ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ

Кафедра АВТОМАТИЗИРОВАННЫХ СИСТЕМ УПРАВЛЕНИЯ

**Лабораторная работа 6**

Дисциплина «Разработка программных приложений и WEB-программирование»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Выполнил:** |  | **Проверил:** |  | |
| Студент | *Корнева Е. С.* | Преподаватель | *Эстрайх И.В.* | |
| Факультет | *АВТФ* |  |  | |
| Направление (специальность) подготовки | *09.03.03 – Прикладная информатика* | Балл: \_\_\_\_\_\_\_\_\_ | |  |
| Группа | *АП-126* | Оценка *\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_* | | |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | | |
| подпись | | подпись | | |
| Дата сдачи: | «\_\_»\_\_\_\_\_\_\_\_\_\_20\_\_г. | Дата защиты: | «\_\_»\_\_\_\_\_\_\_\_\_\_20\_\_г. | |

Новосибирск, 2024

**Использование шаблонов в Django**

**Цель работы:** изучить принципы создания и использования шаблонов в Django, разработать шаблоны для своего проекта.

**Задание:**

Разработать шаблоны для своего сайта. Развить свой проект, используя приёмы, описанные в руководстве: передачу данных в шаблоны, фильтры, теги шаблонов, наследование шаблонов, пользовательские теги шаблонов. Подключить статические файлы, в том числе css.

**Ход выполнения работы:**

1. Введение в шаблоны. Функции render() и render\_to\_string()

Организуем работу с шаблонами. Создадим каталог templates, в котором будем располагать файлы шаблонов.

Создадим в подкаталог templates/abouttrip файл index.html, представляющий главную страницу сайта, а также файл about.html.

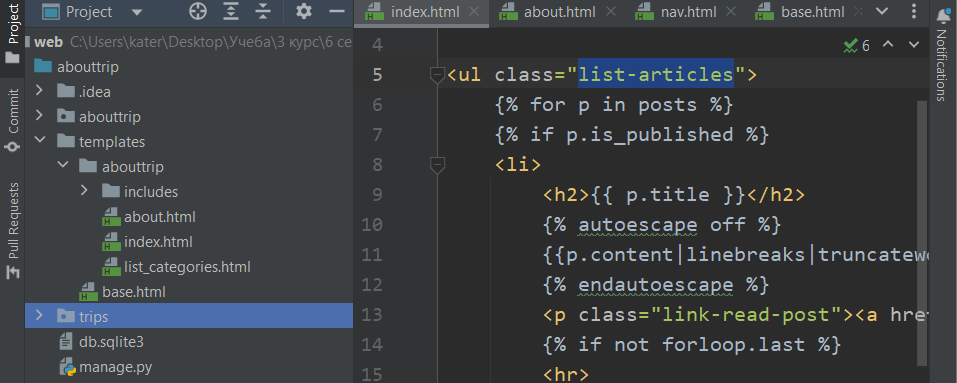


Рис.1. Структура проекта

Создадим функции представления, используя render():



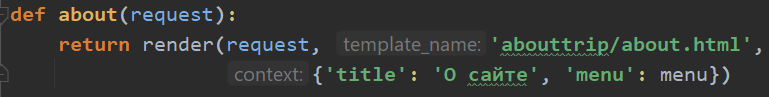
Рис.2. Функции представления

1. Передача данных (переменных) в шаблоны

Сделаем так, чтобы на каждой странице был свой заголовок, переданный ей через параметр title. Это можно сделать следующим образом. В файлах index.html и about.html укажем переменную title:

*<h1>{{title}}</h1>*

А в функциях представлений нужно передать параметр title соответствующему шаблону. Сделаем это с помощью словарей. В нашем случае необходимо определить словарь с ключом title и значением, которое будет подставляться вместо этого ключа:



Тогда при переходе к странице увидим вместо {{title}} строку «О сайте». Вот в этом и есть роль шаблонов: они описывают структуру страницы, а ее наполнение происходит динамически в самой программе.

1. Стандартные шаблонные фильтры

Данные в шаблонах можно преобразовывать, используя фильтры.

Так, для вывода контента на сайте мы используем следующие фильтры:

для того, чтобы выводить не более 50 слов используем следующий фильтр:

*{{p.content|truncatewords:50}}*

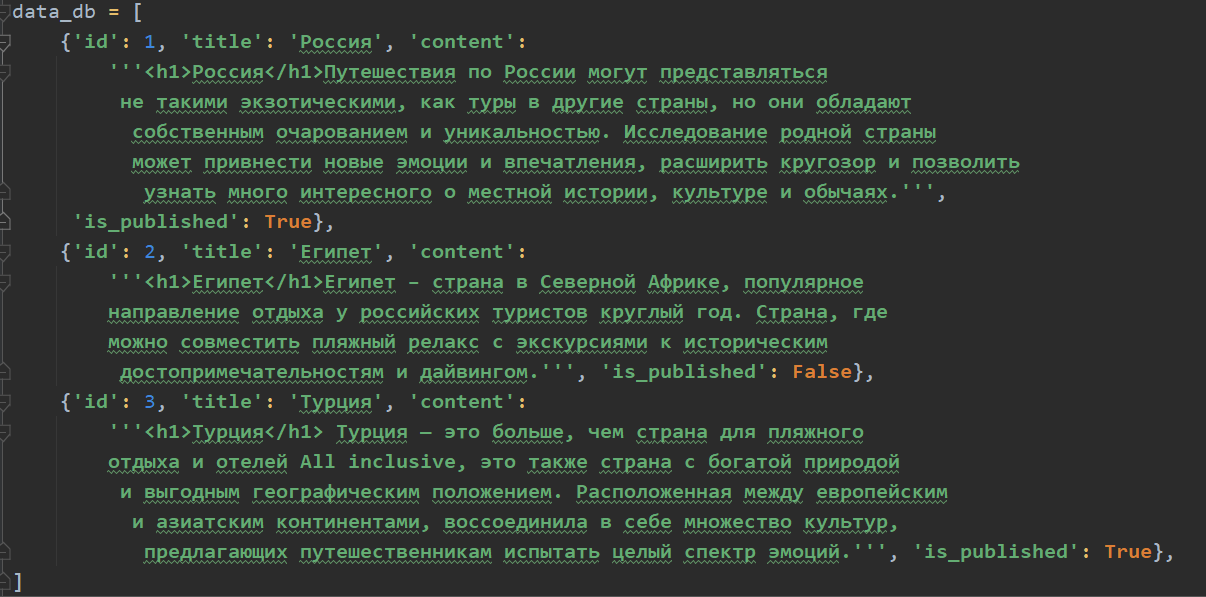
Затем мы сделали так, чтобы каждая строка описания выводилась с новой строки в окне браузера. Для этого добавили еще один фильтр: *{{p.content|linebreaks|truncatewords:50}}*

В ходе работы мы познакомились со многими фильтрами и несколько из них использовали в своем проекте. В заключение отметим, что все эти фильтры можно использовать не только в шаблонах, но и непосредственно в программе на языке Python.

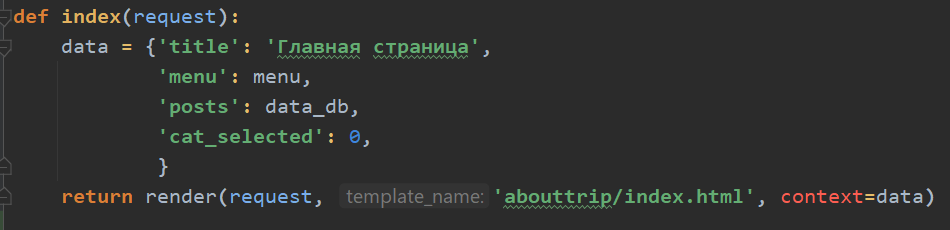
1. Теги шаблонов. Теги if и for

Чтобы было понятно, что собой представляют шаблонные теги и для чего они нужны, рассмотрим два очень распространенных тега if и for. Начнем с тега for.

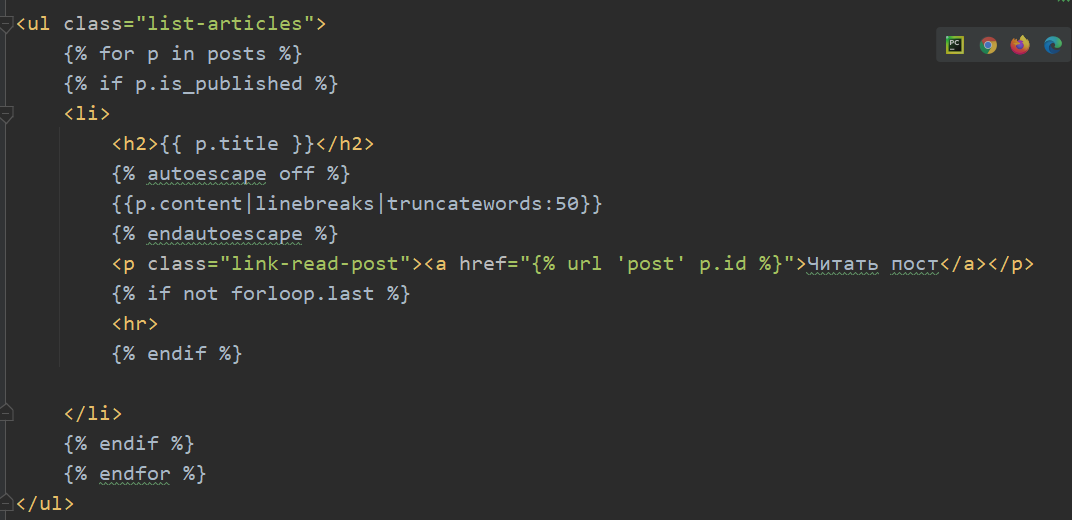
Предположим, что у нас имеется список из данных по странам:



В функции представления index сформируем общие данные и передадим их в шаблон index.html:



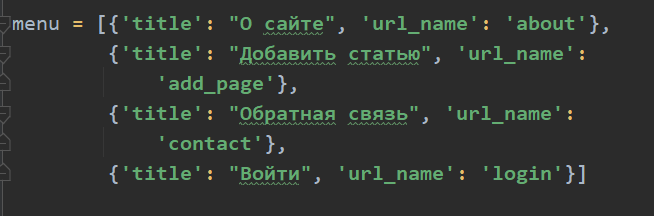
В шаблоне будет доступна коллекция posts из набора публикуемых статей. Наша задача отобразить этот список. Сделаем это следующим образом (в шаблоне index.html), используя тег for, а также if, который позволяет проверять условия и работает подобно оператору if языка Python:



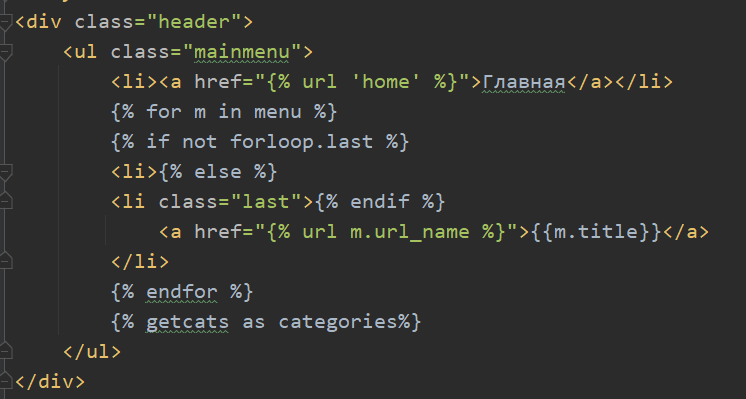
Шаблонные теги for и if применять достаточно просто, и работают они аналогично операторам for и if языка Python

1. Шаблонный тег url

Изменим коллекцию menu, добавив в нее информацию об именах маршрутов (в файле views.py):



А в файле index.html выполним отображение меню с ссылками:



В результате увидим список из ссылок пунктов главного меню сайта.

1. Наследование шаблонов

Дублирование информации в разных шаблонах нарушает известный принцип:

DRY – don’t repeat yourself (не повторяйся).

Для устранения этого недостатка, обычно создается базовый (общий) шаблон, содержащий общие элементы типовых страниц сайта, а уже потом он расширяется (дополняется деталями) шаблонами отдельных страниц. Такое расширение называется наследованием шаблонов.

Создадим файл base.html, в который вынесем общие элементы страниц сайта. Для изменяемой части пропишем специальный шаблонный тег block:

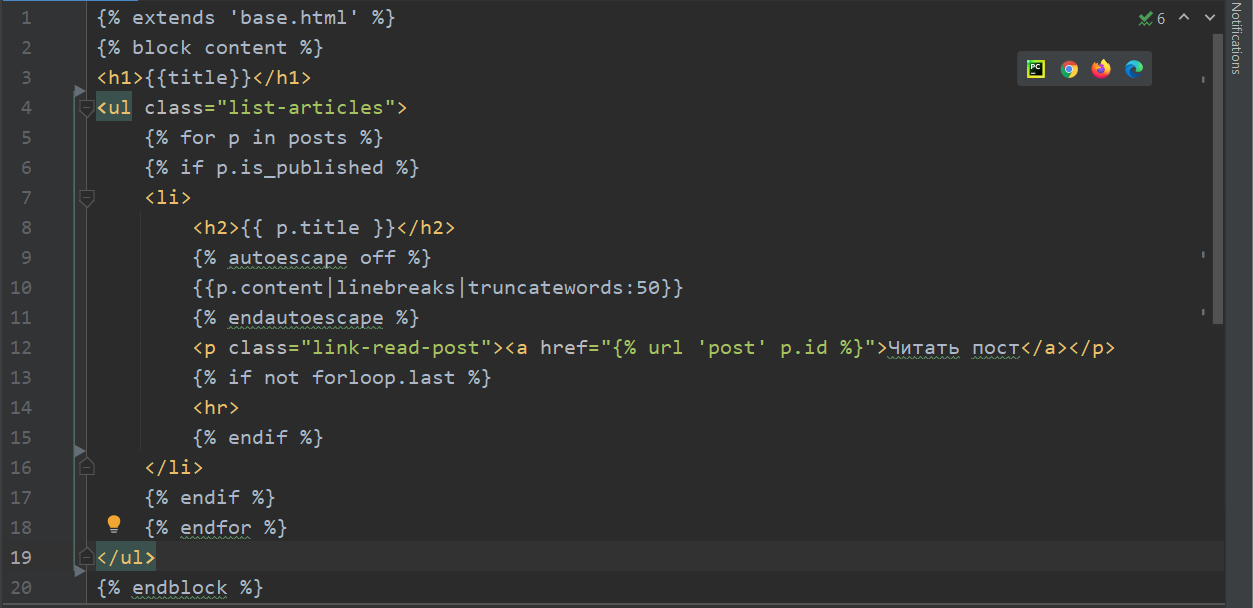
*{% block content %} {% endblock %}*

Здесь content – это название блока, вместо которого будет подставляться содержимое из наследуемых шаблонов.

Далее расширим его для шаблонов index.html и about.html. Сначала перейдем в файл index.html и самой первой строкой пропишем еще один шаблонный тег extends:

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

А всю информацию разместим в блоке content базового шаблона.

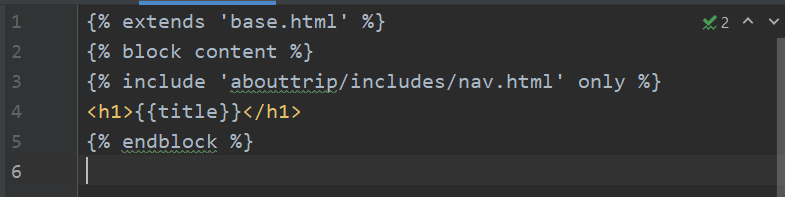


В результате мы ушли от дублирования в отдельных шаблонах и можем достаточно просто определять страницы сайта, просто расширяя базовый шаблон base.html.

1. Тег include

Помимо расширения шаблонов можно еще делать включение одного шаблона в другой, для этого применяется тег include.

Вначале мы создадим подкаталог includes в папке templates/abouttrip нашего проекта для лучшей организации структуры файлов шаблонов. Внутри каталога includes поместим файл nav.html и скопируем в него тег nav. А в шаблоне about.html подключим этот файл с помощью тега include следующим образом:



В итоге получаем довольно гибкий инструмент включения фрагментов в наши шаблоны.

1. Подключение статических файлов

Cоздадим папку static в нашем приложении abouttrip и, также как и для шаблонов, укажем в ней вложенный каталог abouttrip, чтобы не было конфликтов имен между статическими файлами разных приложений.

В этом последнем подкаталоге уже будем размещать файлы css – для файлов CSS; js – для файлов JavaScript; images – для общих файлов изображений и так далее. Создадим подкаталог css для файла стилей нашего сайта. В нем поместим файл styles.css, который нужно подготовить заранее, Также создадим подкаталог images и скопируем в него все необходимые изображения для базового оформления сайта.

Теперь мы можем использовать эти внешние файлы в шаблонах нашего приложения. Для этого есть специальный тег static, который сначала подключается в шаблоне (base.html):

*{% load static %}*

Lля формирования URL к тому или иному статическому файлу, используется специальная конструкция, например, для подключения css-файла в базовом шаблоне base.html, следует прописать:



Далее, возьмем заготовленные html-файлы для базового шаблона страниц (base.html) и главной страницы (index.html).

В результате получим следующую страницу:

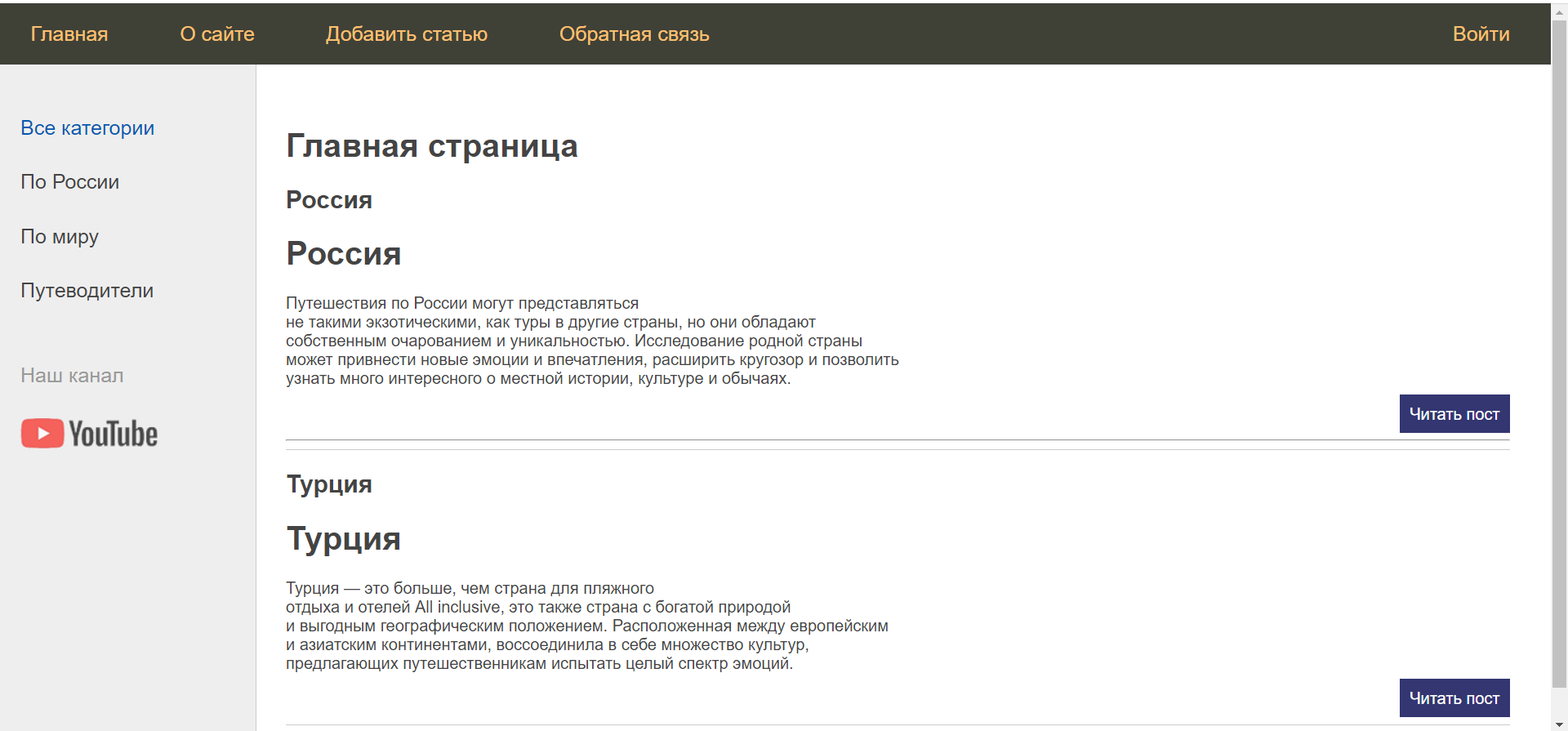


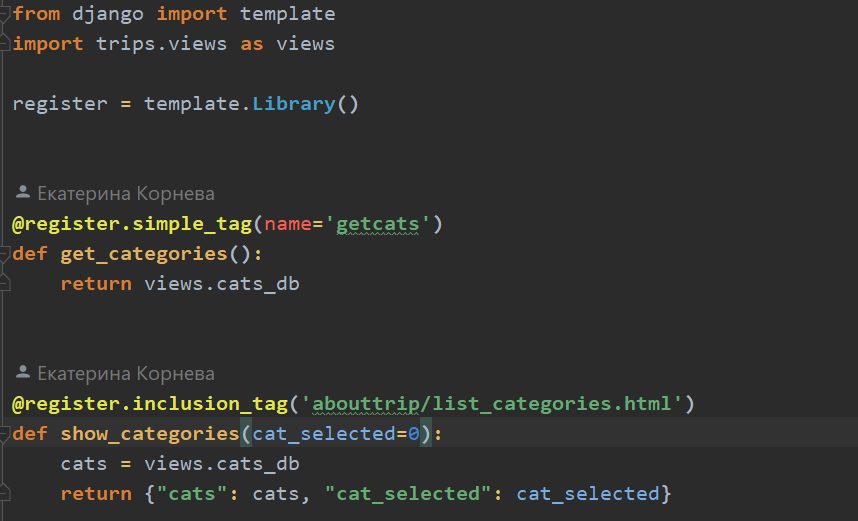
Рис.3. Внешний вид сайта

1. Пользовательские теги шаблонов

Мы можем создавать свои собственные шаблонные теги и использовать их при формировании HTML-страниц. Для этого Django позволяет использовать два вида пользовательских тегов:

• simple tags – простые теги;

• inclusion tags – включающие теги.



**Вывод**

В ходе работы мы изучили принципы создания и использования шаблонов в Django, а также разработали шаблоны для своего проекта.

Ссылка на github: https://github.com/KateMeon/web.git