МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И

МАССОВЫХ КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ

Ордена Трудового Красного Знамени федеральное государственное бюджетное образовательное учреждение высшего образования

«МОСКОВСКИЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

СВЯЗИ И ИНФОРМАТИКИ»

Кафедра «Сети связи и системы коммутации»

Лабораторная работа №2

«Создание веб-страницы с текстом»

по дисциплине «Web-программирование»

Выполнил студент

группы БФИ1901

Шацкий Е.И.

Проверила:

Полянцева К.А.

Москва, 2021

**Содержание**

[1 Задание на лабораторную работу 3](#_Toc87017478)

[2 Ход лабораторной работы. 3](#_Toc87017479)

[2.1 Создание первой web-страницы с простым текстом. 3](#_Toc87017480)

[2.2 Создание первого html-шаблона. 8](#_Toc87017481)

[2.3 Настройка обработки статичных файлов для django. 15](#_Toc87017482)

[Список используемых источников 22](#_Toc87017484)

# **1 Задание на лабораторную работу**

1. Создать веб-страницу с текстом.

2. Изучить основы работы с шаблонами в Django

3. Настроить обработку статичных файлов в Django.

# **2 Ход лабораторной работы.**

## 2.1 Создание первой web-страницы с простым текстом.

Для начала необходимо создать проект Django. Для этого необходимо выполнить команду «django-admin.exe startproject firstwebpage». Это представлено на рисунке 1. Структура, получившегося проекта firstwebpage представлена на рисунке 2 и 3.

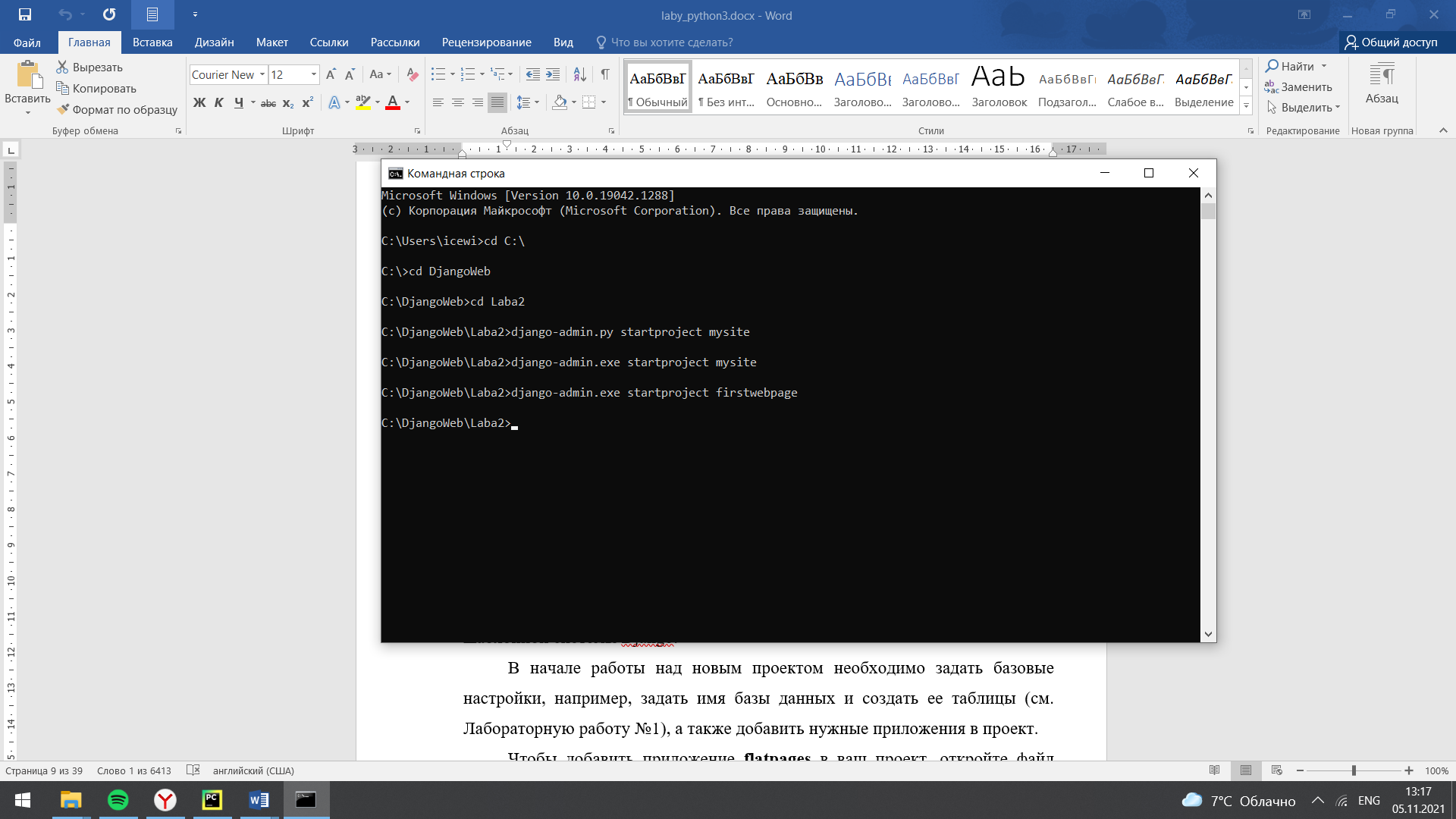


Рисунок 1 – Создание проекта

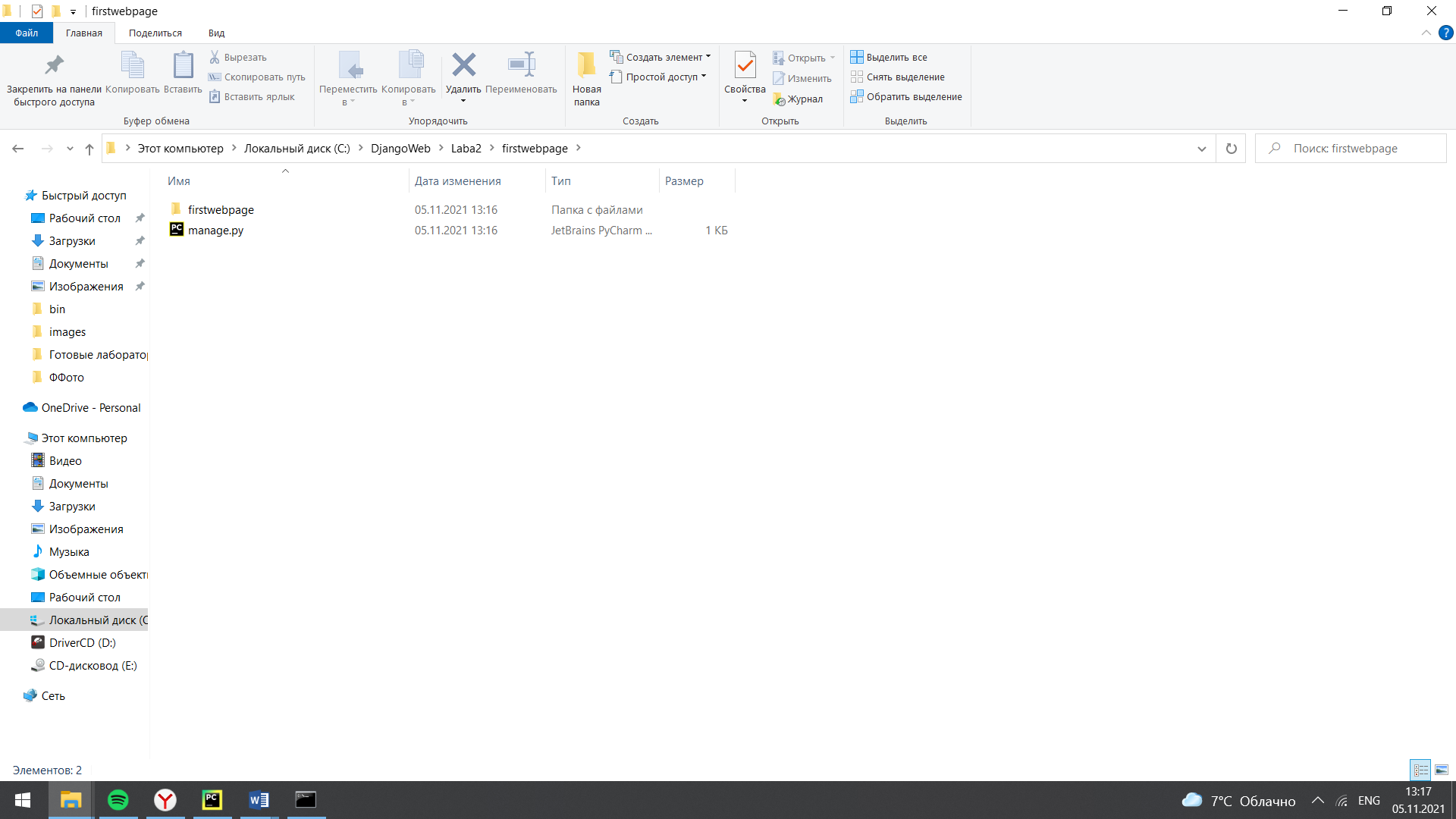


Рисунок 2 – Структура проекта firstwebpage

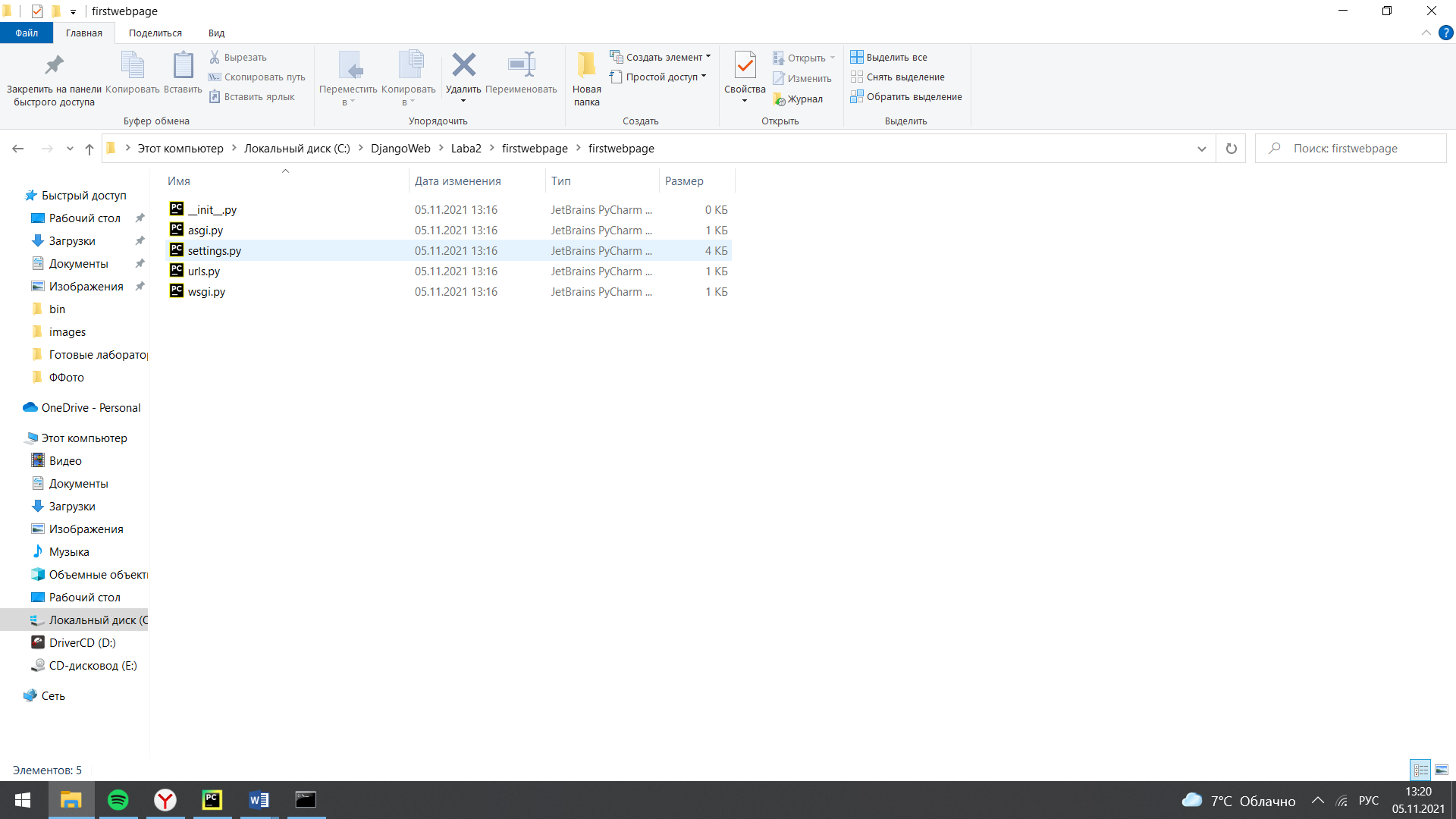


Рисунок 3 – Структура проекта firstwebpage

Далее необходимо подключить новое приложение для проекта. Для этого необходимо выполнить команду «python manage.py startapp flatpages». Приложение flatpages позволяет вам управлять статическими страницами через интерфейс администратора Django и указывать шаблоны для таких страниц с помощью шаблонной системы Django. Это представлено на рисунке 4. Изменения в каталоге проекта показаны на рисунке 5.

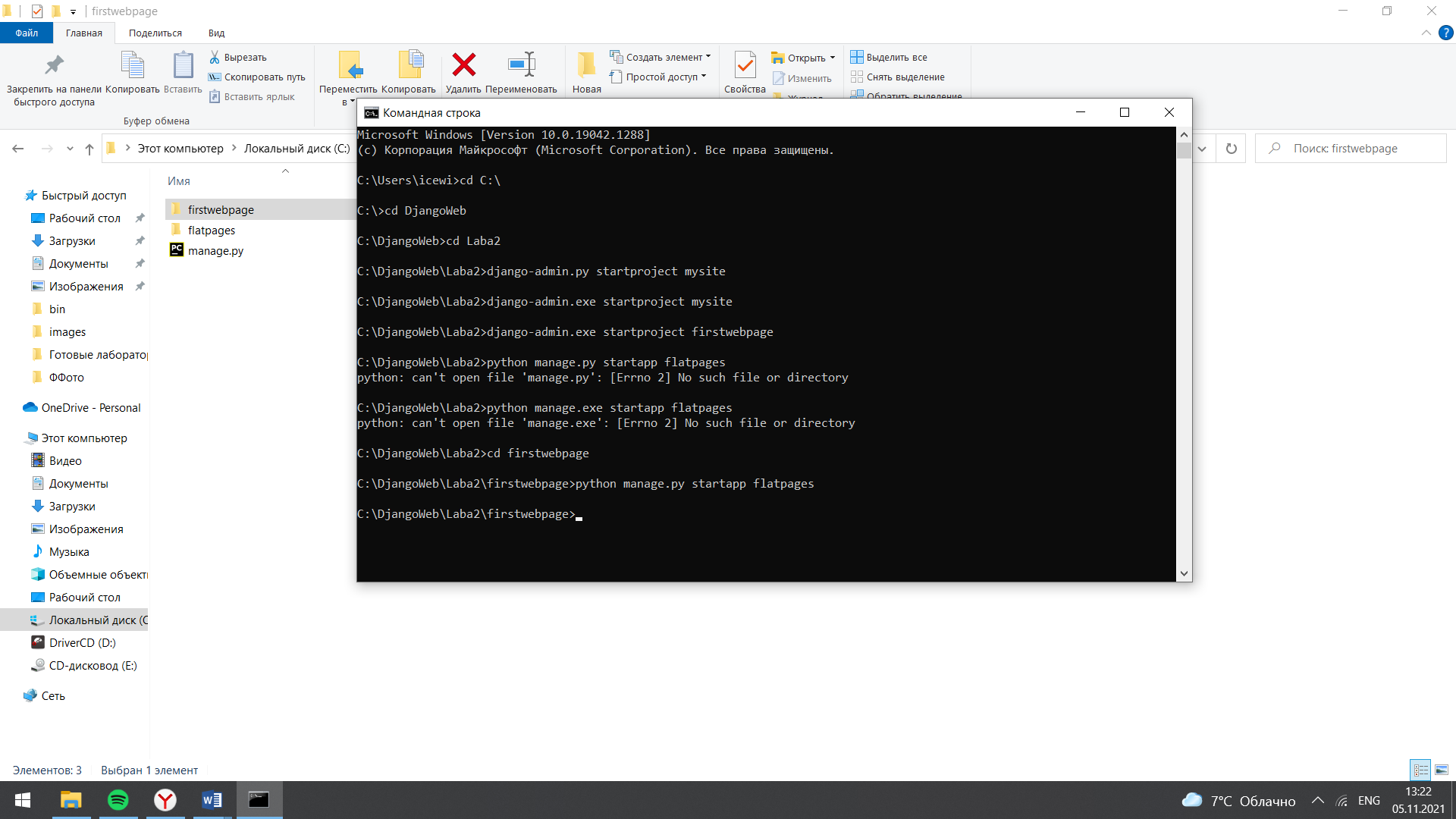


Рисунок 4 – Подключение приложения «flatpages» в проект

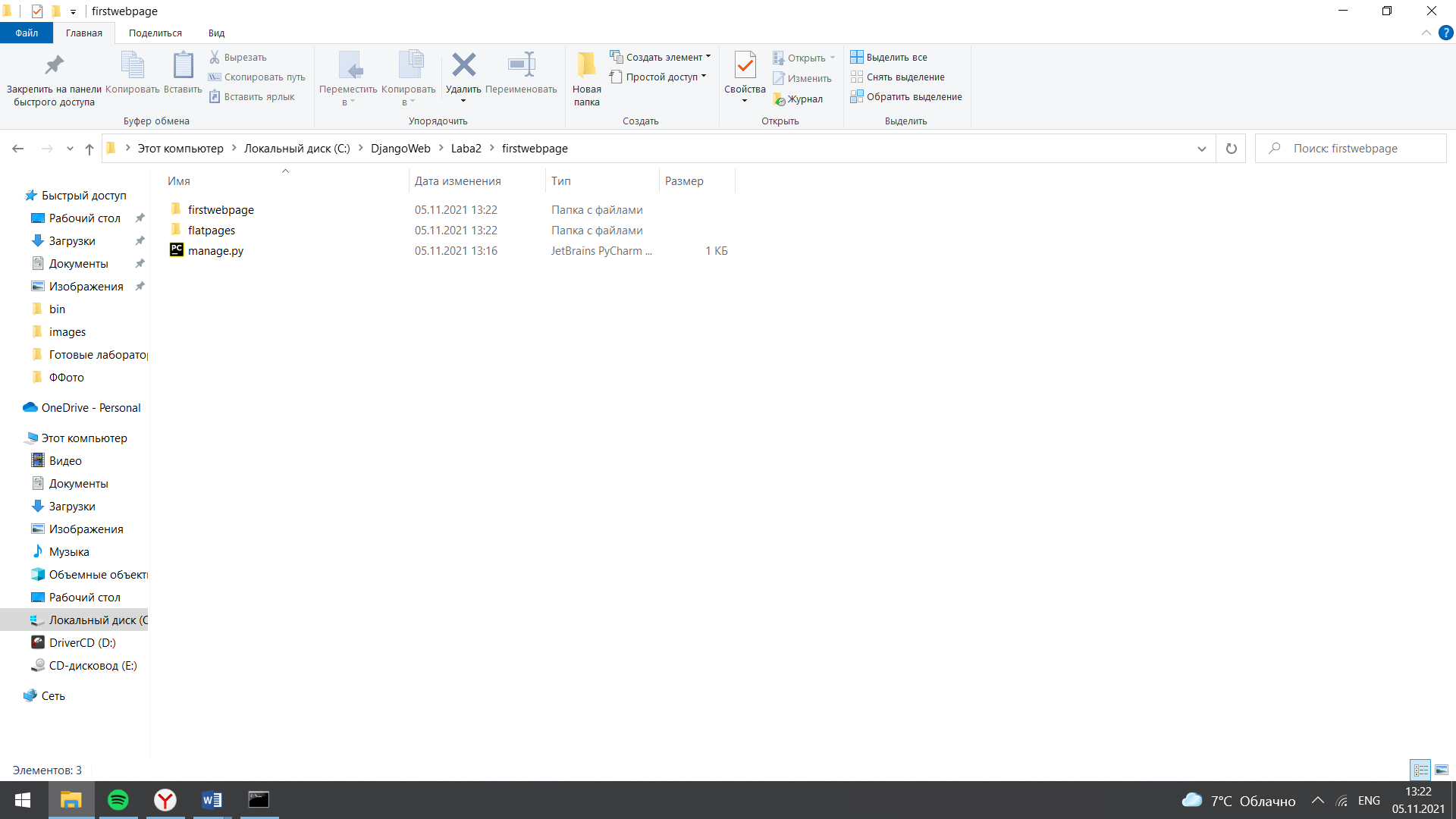


Рисунок 5 – Изменения в каталоге проекта

Затем нам необходимо поменять настройки базы данных на такие же, как и в Лабораторной работе №1. В файле settings.py необходимо изменить параметр базы данных. Переменной DATABASES есть внутренний словарь с ключами ENGINE и NAME. Для параметра NAME установливаю следующее значение: «'NAME': os.path.join(BASE\_DIR, 'shatsky\_db')». Это действие представлено на рисунке 6.

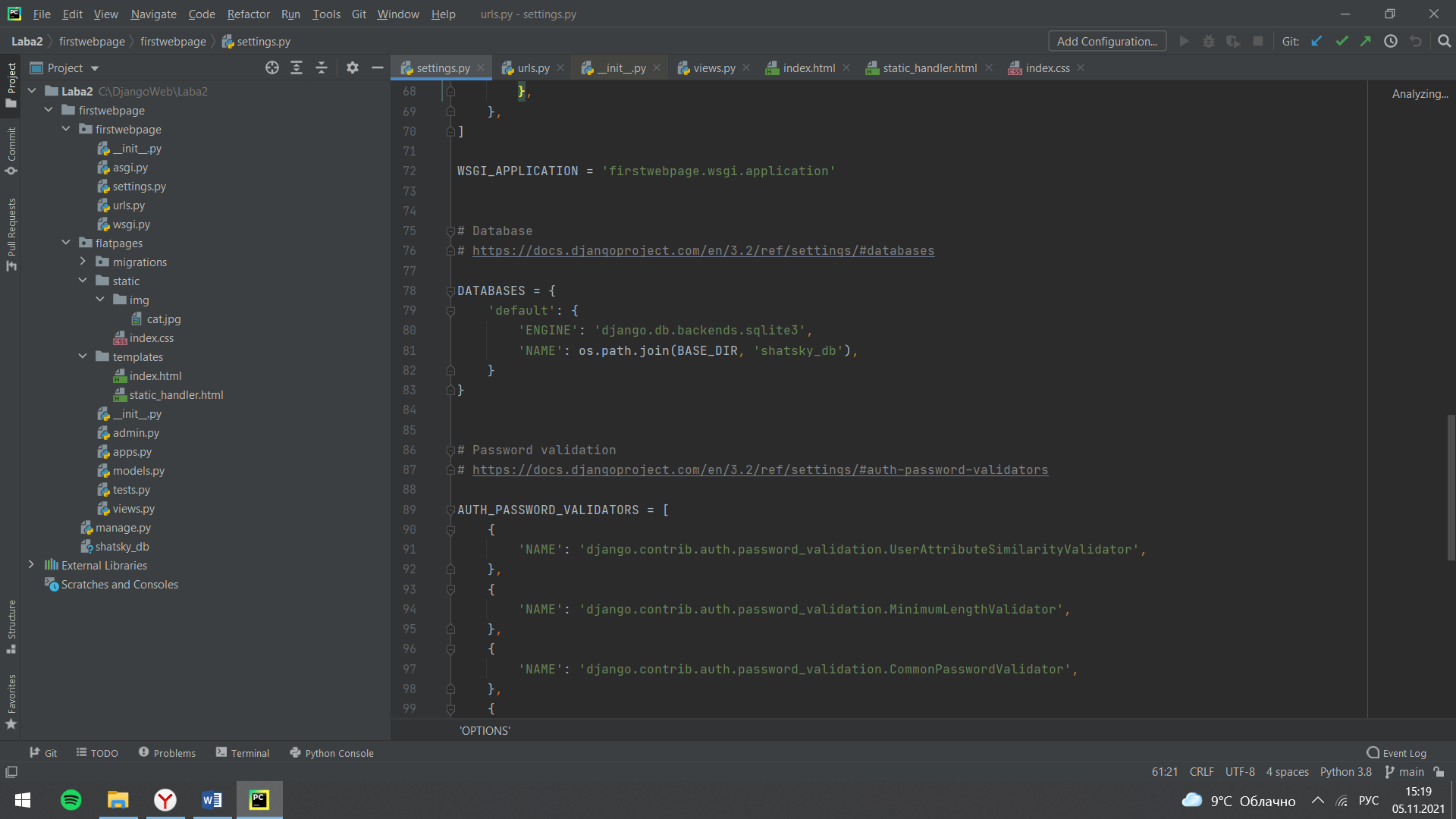


Рисунок 6 – Изменение параметра DATABASES

Далее необходимо добавить приложение «flatpages» в проект. Для этого в файле settings.py в кортеже INSTALLED\_APPS и добавить в конец элемента строку 'flatpages'. Это представлено на рисунке 7.

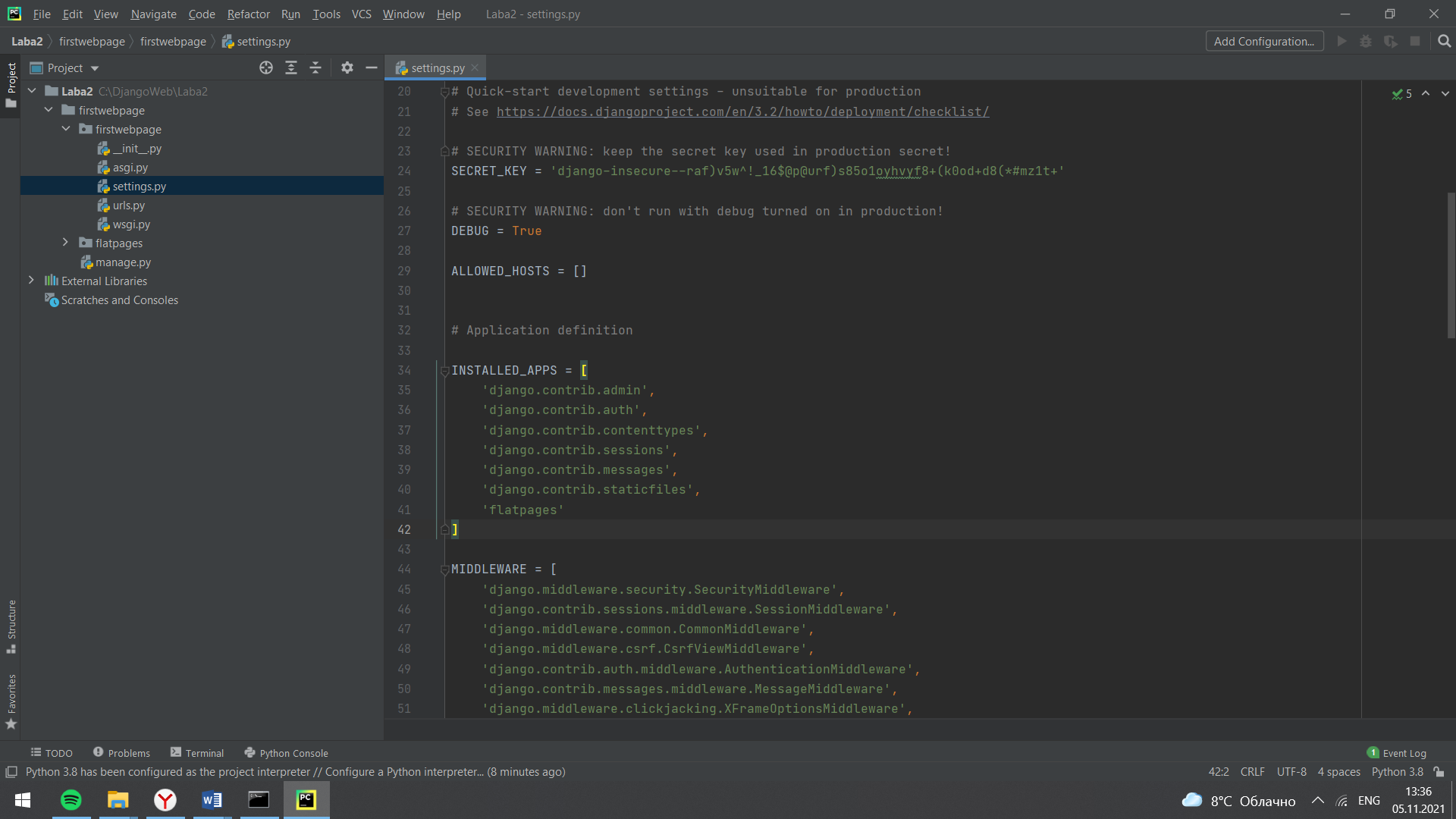


Рисунок 7 – Добавление приложения «flatpages» в проект

Перед началом работы с файлом urls.py необходимо импортировать views нашего приложения с помощью команды «from flatpages import views».

Для будущей страницы создам новый адрес в файле urls.py:

urlpatterns = [

path('admin/', admin.site.urls),

path('', views.home, name='home'),

]

Второй аргумент, передаваемый в функцию, указывает на то, что функция представления, которая будет обрабатывать запросы по этому адресу, находится в файле views и называется home.

Для того, чтобы в будущем при обращении генерировался ответ, необходимо создать функцию home в файле views.py в директории flatpages.

from django.http import HttpResponse

def home(request):

return HttpResponse(u'Hello World!', content\_type="text/plain")

На рисунке 8 представлено содержимое файла urls.py. На рисунке 9 представлено содержимое файла view.py.



Рисунок 8 – Файл urls.py

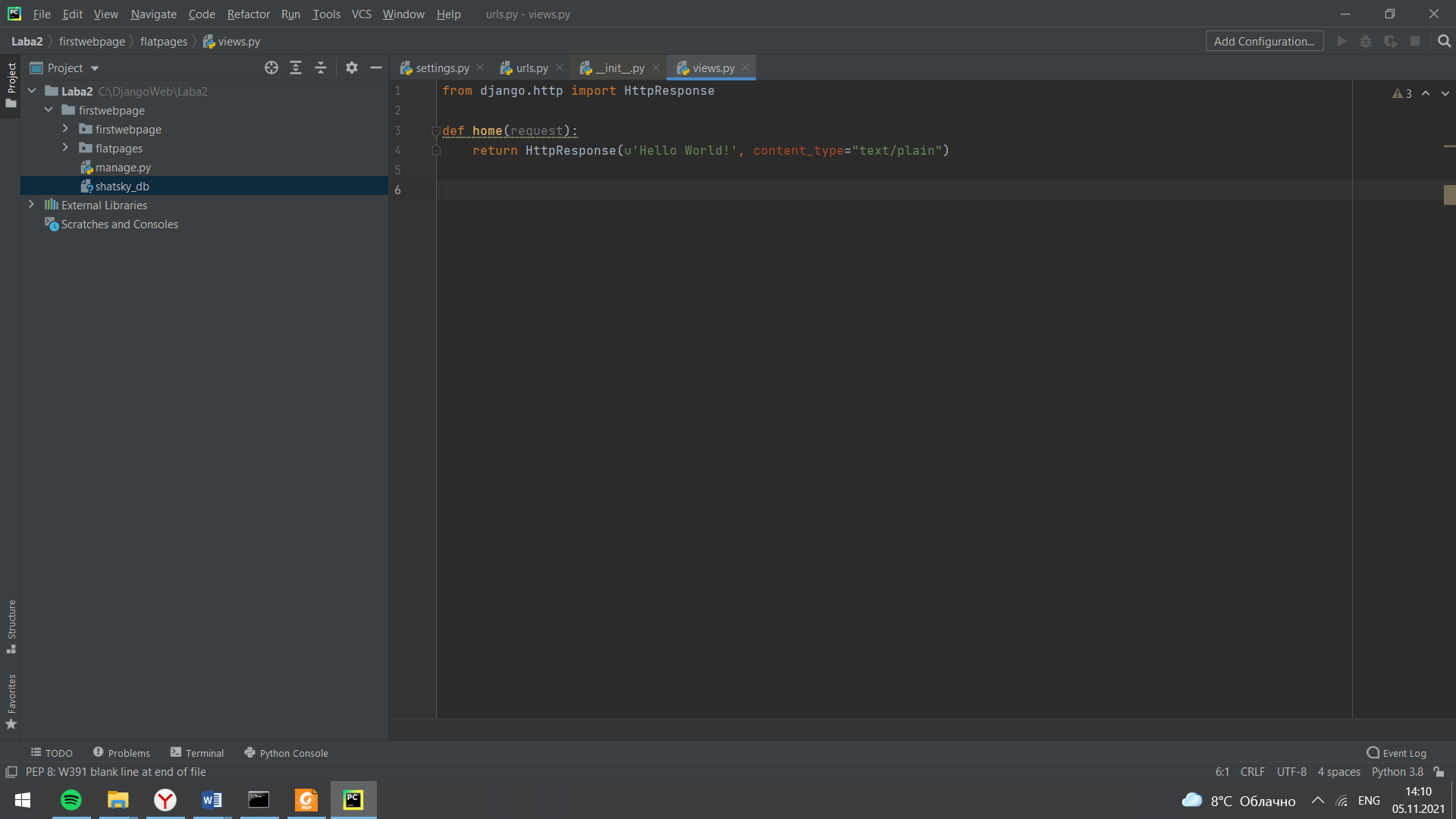


Рисунок 9 – Файл view.py

Далее запустим локальный сервер с помощью команды python manage.py runserver и проверим результаты перейдя по адресу «http://127.0.0.1:8000/». Результаты представлены на рисунке 10.

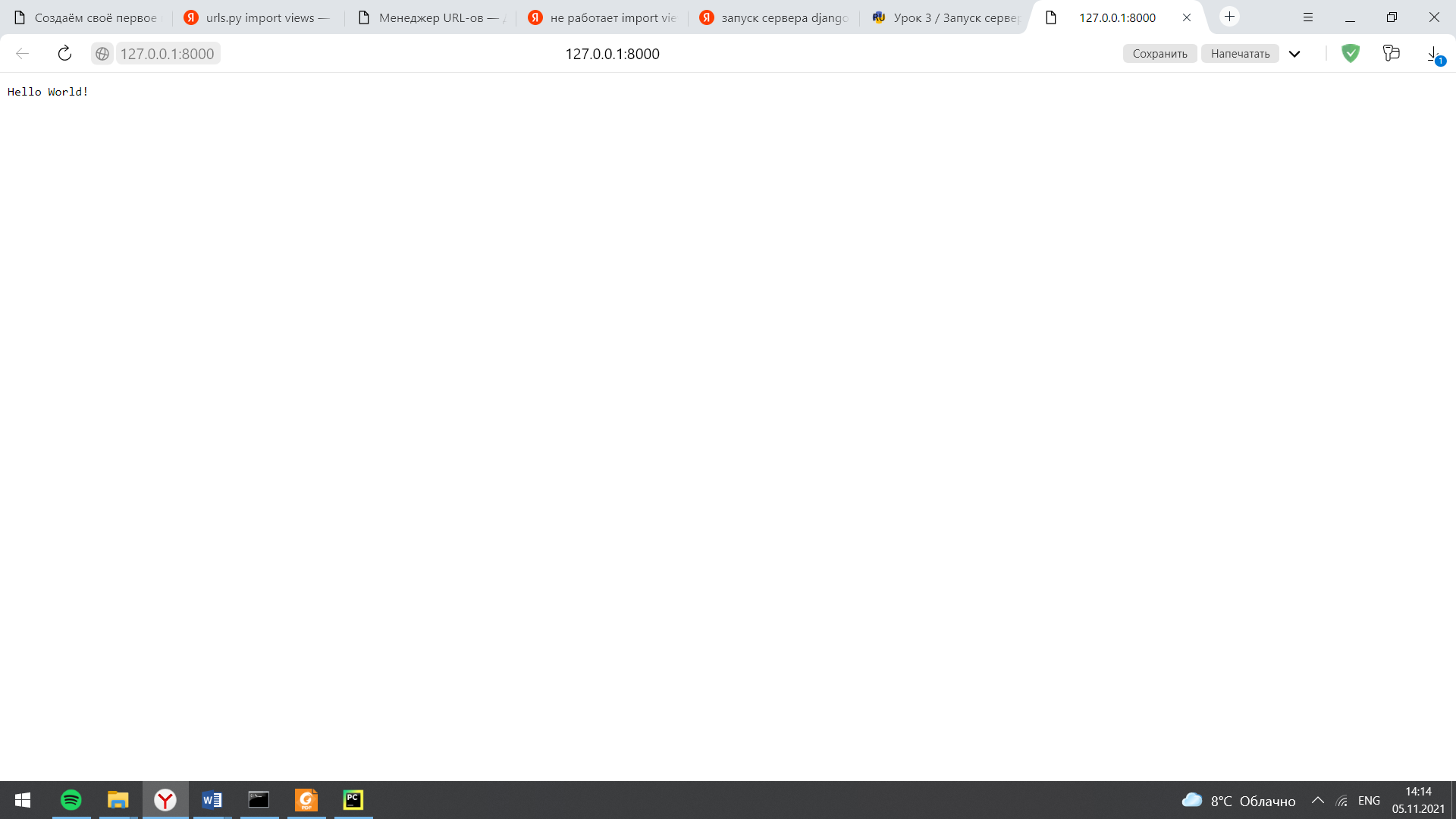


Рисунок 10 – Результаты работы сервера

Затем по плану лабораторной работы необходимо сделать так, что по адресу http://127.0.0.1:8000/hello/ возвращался тот же самый текст. Для этого в файле urls.py в «urlpatterns» добавим «path('hello/', views.home, name='hello'),». После этого после перехода по адресу «http://127.0.0.1:8000/hello/» у нас будет выведен тот же текст что и при переходе по адресу «http://127.0.0.1:8000/». Результат этого действия показан на рисунке 11.

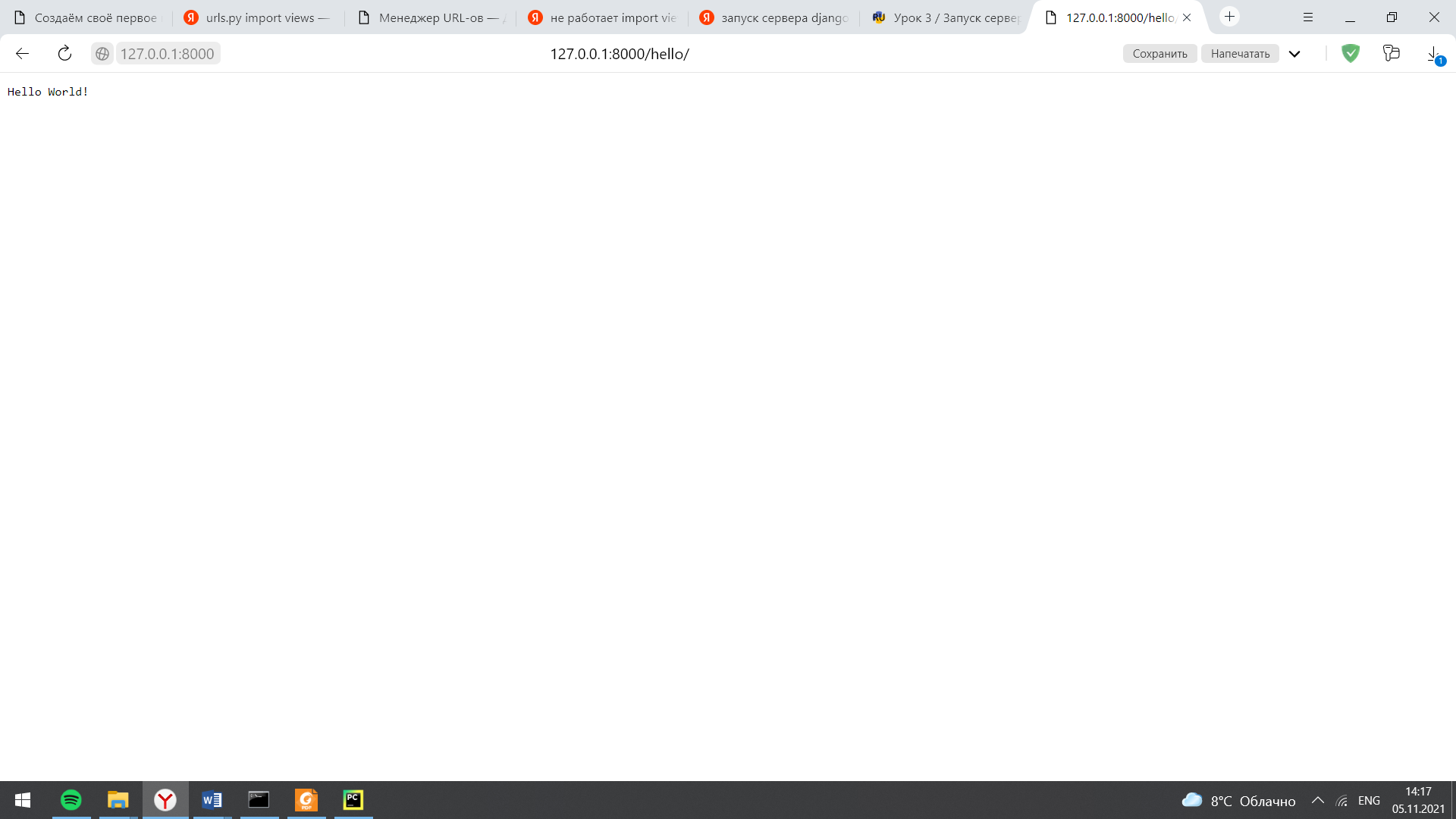


Рисунок 11 – Hello World после перехода по ссылке http://127.0.0.1:8000/hello

Далее необходимо убрать указание типа возвращаемого ответа классу HttpResponsе и сравнить результаты. На рисунке 12 показан, измененный ответ классу HttpResponsе, а результат после удаления типа возвращаемого ответа продемонстрирован на рисунке 13.

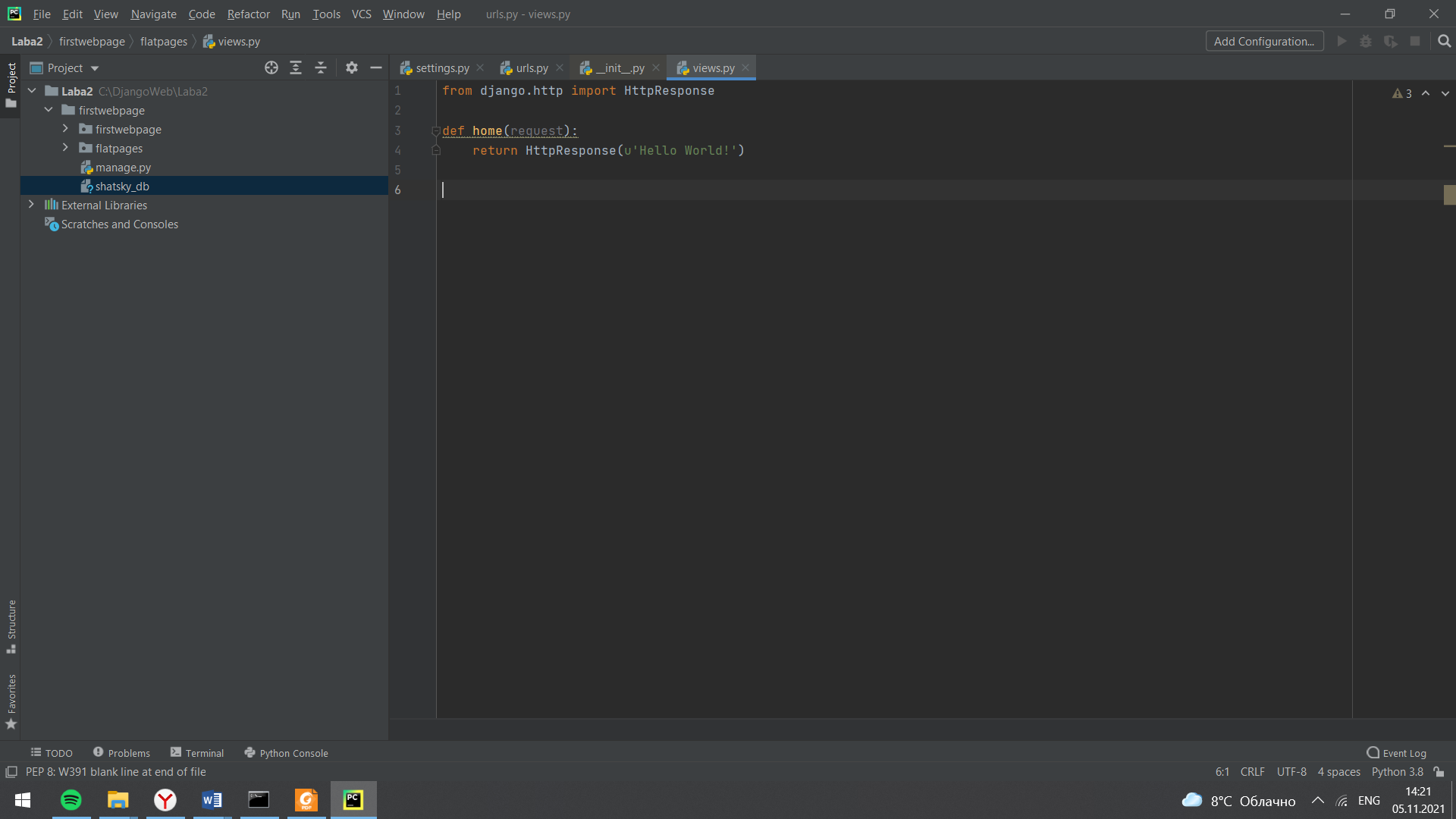


Рисунок 12 - ответ классу HttpResponsе

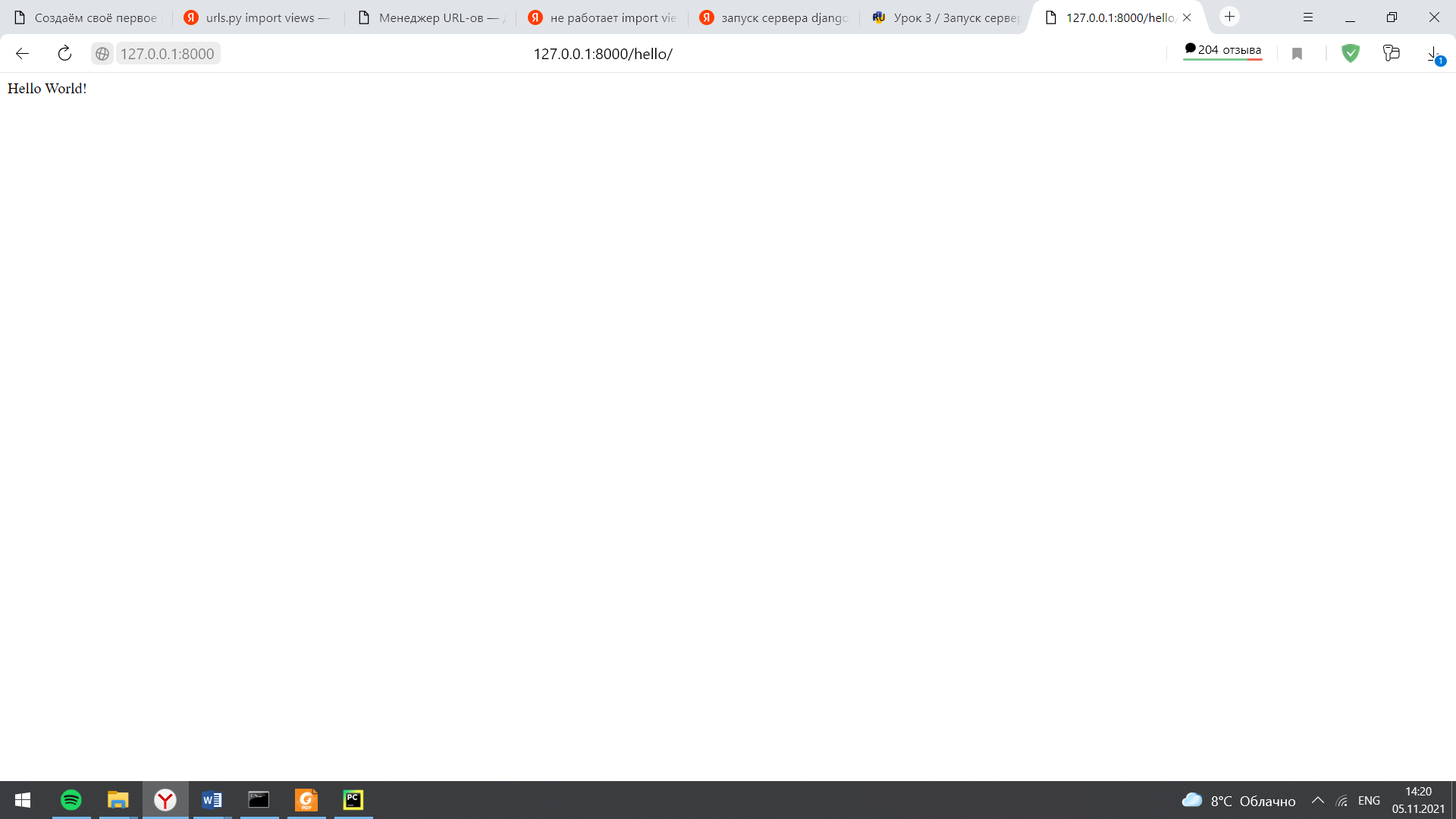


Рисунок 13 - Результат после удаления типа возвращаемого ответа

Можно заметить что после удаления указания типа возвращаемого ответа классу HttpResponsе, шрифт и размер текста поменялся.

**2.2 Создание первого html-шаблона.**

Создам папку templates в директории flatpages. Затем, в папке templates создайте файл index.html с кодом который представлен в Листинге 1.

Листинг 1 – файл index.html

<!DOCTYPE html>

<html>

<head>

<title>Привет, Мир!</title>

</head>

<body>

<h1>Привет, Мир!</h1>

<h2>Это учебный сайт, с его помощью будут изучены технологии

python/django, html/css.</h2>

<h3>Как видите, здесь используются заголовки различных

уровней.</h3>

<p>Здесь есть маркированный список:</p>

<h4>

<ul>

<li>Элемент 1;</li>

<li>элемент 2;</li>

<li>элемент 3;</li>

<li>последний элемент.</li>

</ul>

</h4>

<p>И нумерованный список:</p>

<h4>

<ol>

<li>Элемент 1;</li>

<li>элемент 2;</li>

<li>элемент 3;</li>

<li>последний элемент.</li>

</ol>

</h4>

<p>И даже таблица:</p>

<table style="border: none">

<thead>

<tr>

<th>Столбик 1</th>

<th>Столбик 2</th>

<th>Столбик 3</th>

<th>Столбик 4</th>

</tr>

</thead>

<tr>

<td>Строка 1 Столбец 1</td>

<td>Строка 1 Столбец 2</td>

<td>Строка 1 Столбец 3</td>

<td>Строка 1 Столбец 4</td>

</tr>

<tr>

<td>Строка 2 Столбец 1</td>

<td>Строка 2 Столбец 2</td>

<td>Строка 2 Столбец 3</td>

<td>Строка 2 Столбец 4</td>

</tr>

<tr>

<td>Строка 3 Столбец 1</td>

<td>Строка 3 Столбец 2</td>

<td>Строка 3 Столбец 3</td>

<td>Строка 3 Столбец 4</td>

</tr>

<tr>

<td>Строка 4 Столбец 1</td>

<td>Строка 4 Столбец 2</td>

<td>Строка 4 Столбец 3</td>

<td>Строка 4 Столбец 4</td>

</tr>

<tr>

<td>Строка 5 Столбец 1</td>

<td>Строка 5 Столбец 2</td>

<td>Строка 5 Столбец 3</td>

<td>Строка 5 Столбец 4</td>

</tr>

</table>

</body>

</html

Далее, данный файл необходимо подключить. к функции-представлению home, чтобы вместо простого текстового ответа приходил html-документ. Для этого в файле views.py, который находится в директории flatpages, добавьте операции импортирования:

from django.shortcuts import render

from django import template

А функцию-представление home изменим следующим образом:

def home(request):

return render(request, 'templates/index.html')

Также, для того, чтобы файл index.html был найден в директории templates, необходимо в файле settings.py изменить поле DIRS кортеже TEMPLATES. Поле DIRS должно содержать адрес директории, в которой располагается файл index.html, например:

'DIRS': [os.path.join(BASE\_DIR, 'flatpages'),],

Теперь перейдём на страницу с адресом <http://127.0.0.1:8000/>, и посмотрим результаты. На рисунке 14 представлена страница с результатами вышеописанных операций.

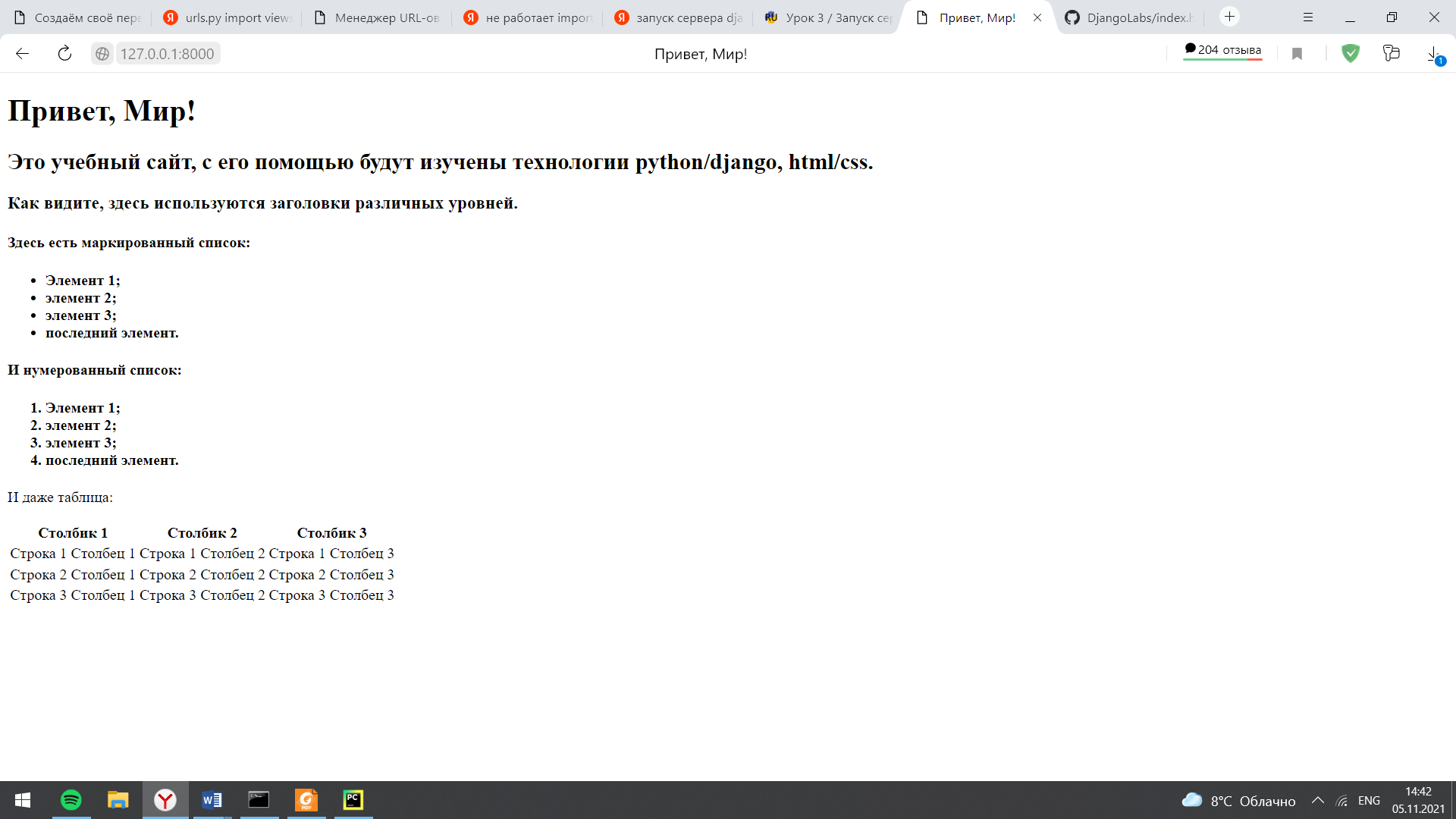


Рисунок 14 – Результаты

Далее необходимо добавить к созданной таблице две строки и два столбца. Для добавления двух строк необходимо воспользоваться тэгом <td> </td>. Для создания строки необходимо воспользоваться тэгом <tr> </tr>. На рисунке 10 представлен результат добавления двух строк и двух столбцов.

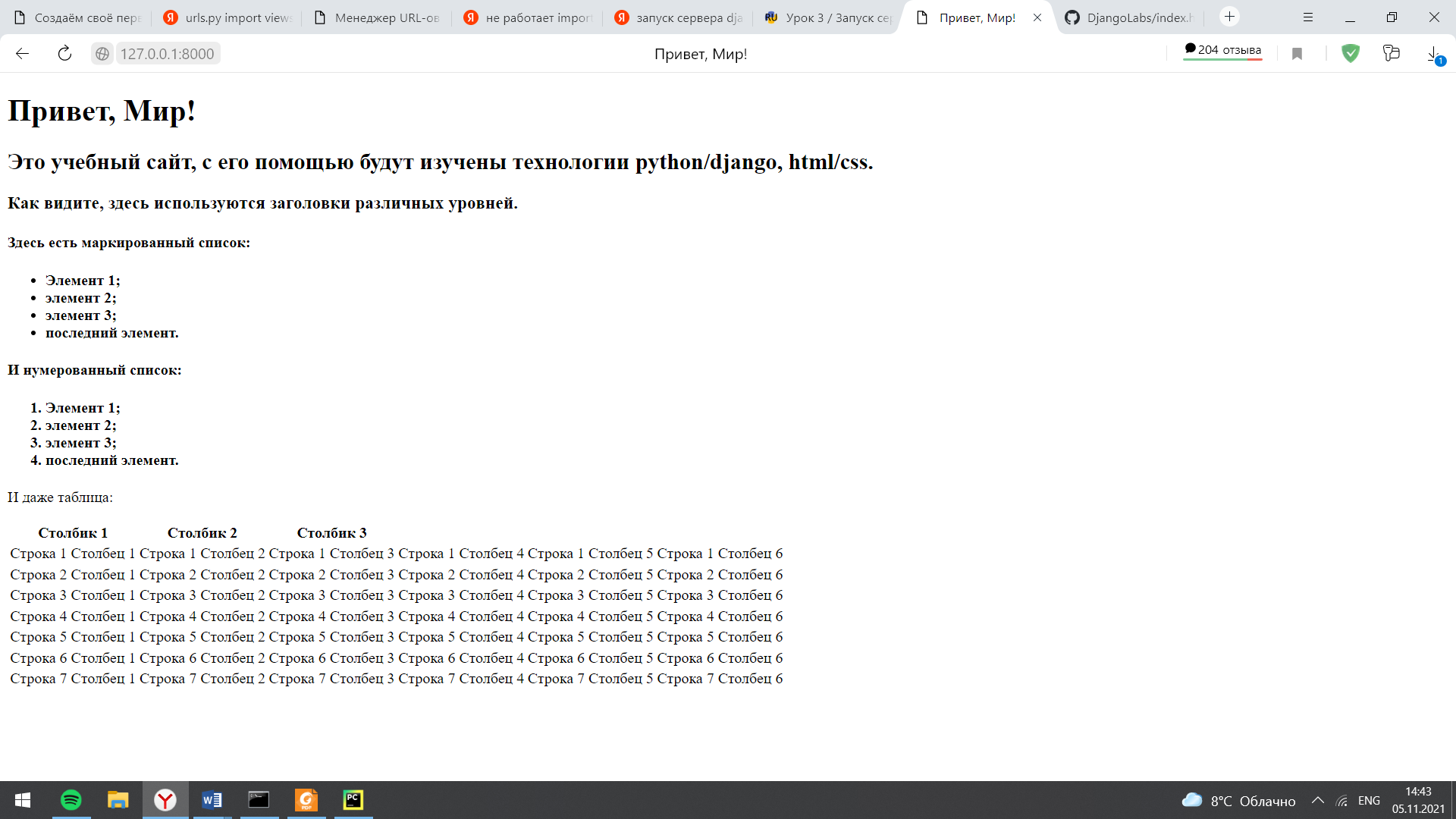


Рисунок 15 - Результат добавления двух строк и двух столбцов

Затем необходимо добавить границы для таблицы. Для этого добавим тэгу <table> атрибут border = "1". Результат представлен ан рисунке 16.

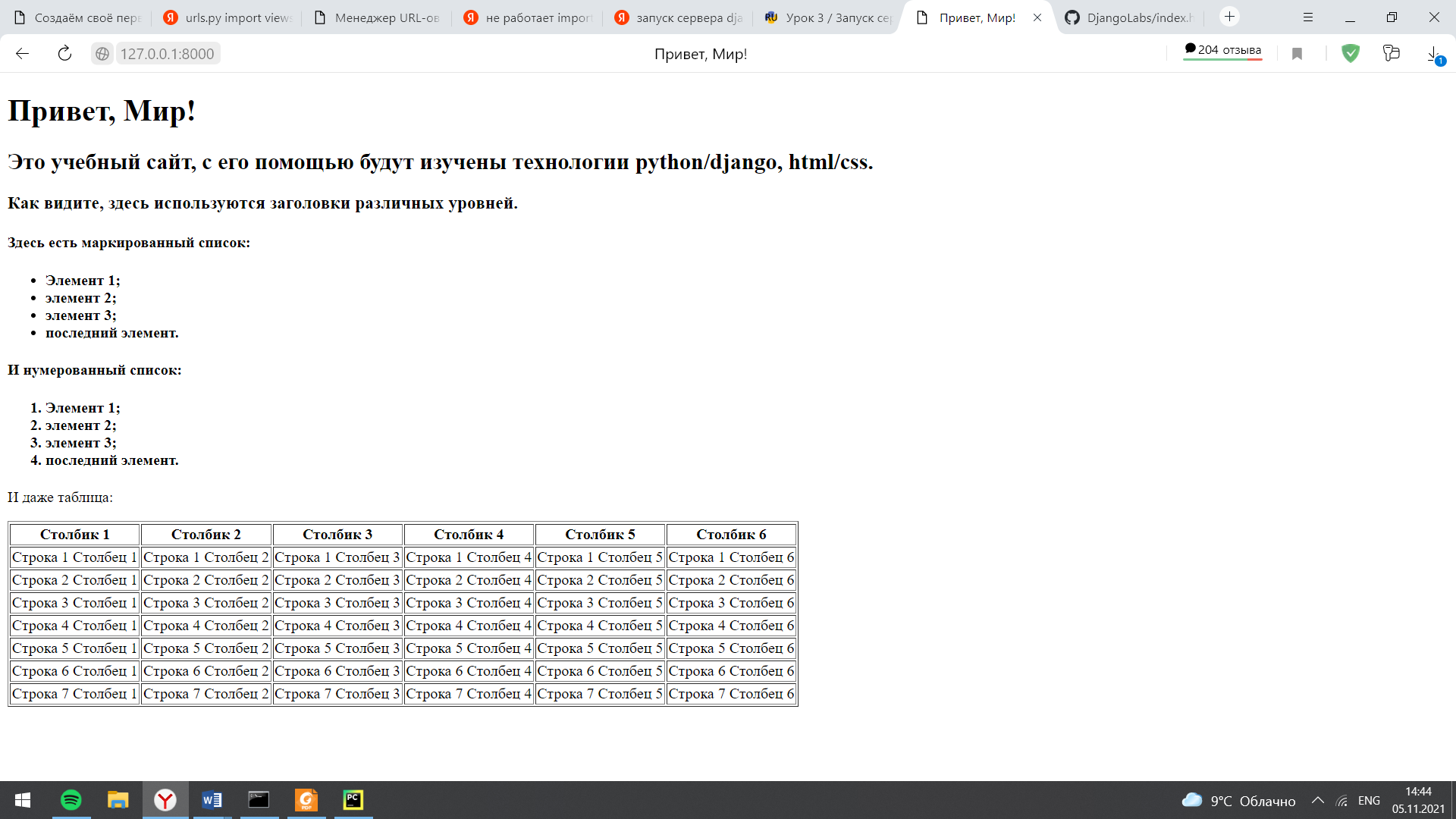


Рисунок 16 – Добавление границ к таблице

Далее необходимо сделать заголовки списков, заголовками четвертого уровня. Для этого заменим тэг <p> на тэг <h4>. Результат представлен на рисунке 17.

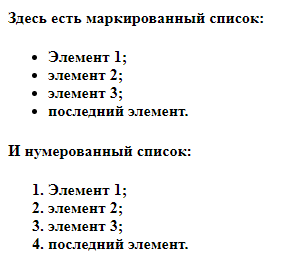


Рисунок 17 – Изменение заголовков списка

Для выполнения последней части лабораторной работы необходимо создать идентичный шаблон. Для этого просто скопируем «index.html» в папку templates и переименуем в «static\_handler.html».

## **2.3 Настройка обработки статичных файлов для django.**

Для начала необходимо создать папку static в директории flatpages, а в ней файл index.css, код которого представлен на Листинге 2.

Листинг 2 - Файл index.css

body {  
background: #1abc9c;  
font-family: Tahoma, Arial, sans-serif;  
color: #333;  
}  
table {  
border-collapse: collapse;  
}  
p, h4 {  
font-size: 22px;  
margin-bottom: 0;  
}  
ul, ol {  
margin: 0  
}  
table tr td {  
padding: 5px  
}

После этого в файл страницы static\_handler.html во внутрь тега <head> вставьте тег подключения css-скрипта:

<link rel="stylesheet" href="{{ STATIC\_URL }}/static/index.css">

На рисунке 18 представлен сайт, с подключением index.css.

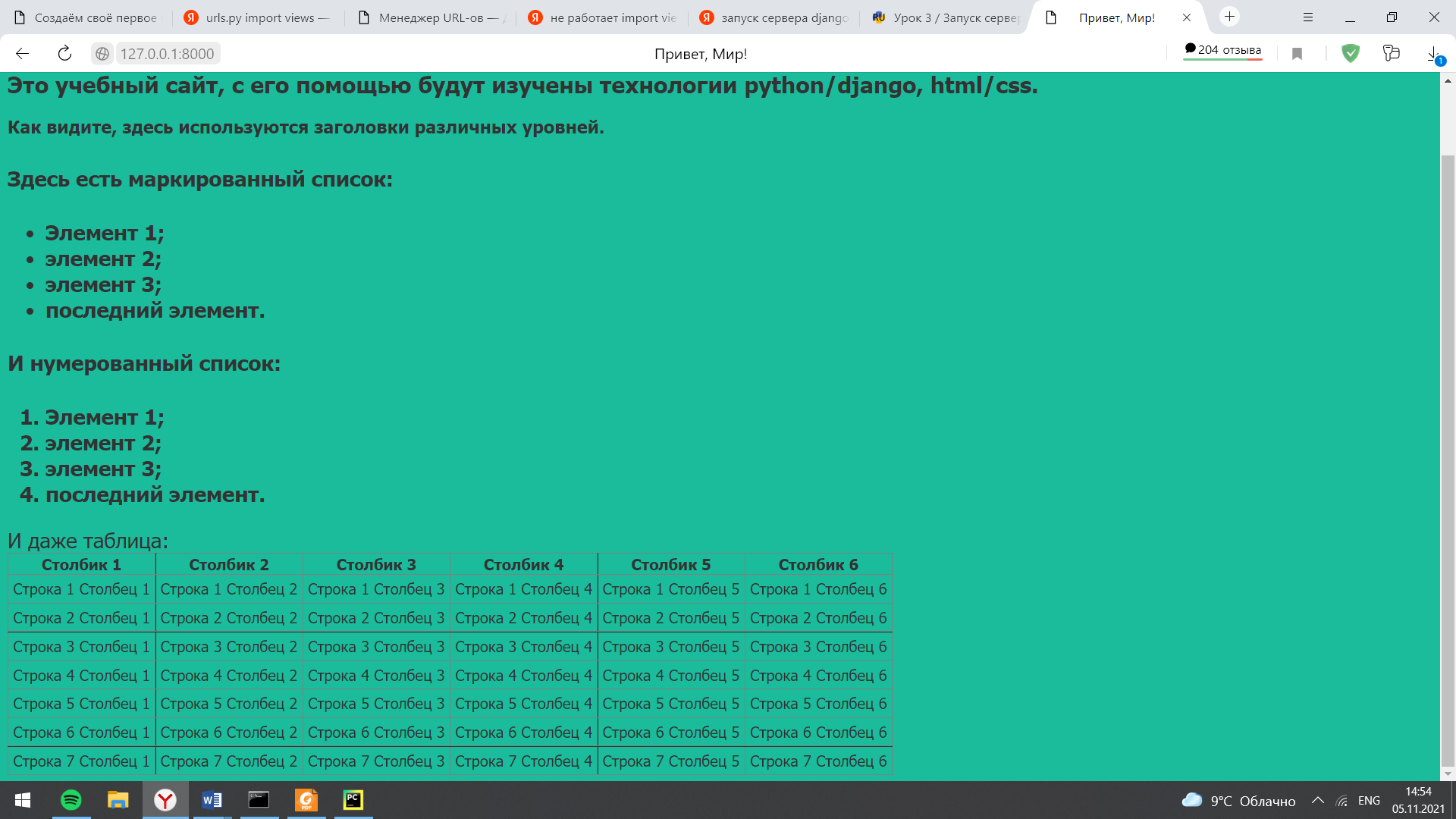


Рисунок 18 – Сайт с подключенными стилями.

Далее необходимо установить для заголовка первого уровня шрифт с засечками. Для этого в файле index.css пропишем:

h1{  
 font-family: Serif;  
 }

Serif означает то, что шрифт с засечками. На рисунке 19 представлен измененный заголовок первого уровня.



Рисунок 19 – Измененный заголовок первого уровня.

Затем необходимо добавить картинку и сделать её высотой 30 пикселей. Для этого поместим картинку в папку static b порпишем «<img src="{{ STATIC\_URL }}img/cat.jpg"/>» после первого тэга <h3> . Результат размещения картинки на сайт представлен ан рисунке 20.



Рисунок 20 – Картинка на сайте

Далее необходимо установить высоту картинки в 30 px, для этого в файле index.css прописать:

img {

height: 30px;

width: auto;

}

Для корректного отображения, было прописано wight: auto что бы картинка не было растянута по ширине. Результат представлен на рисунке 21.

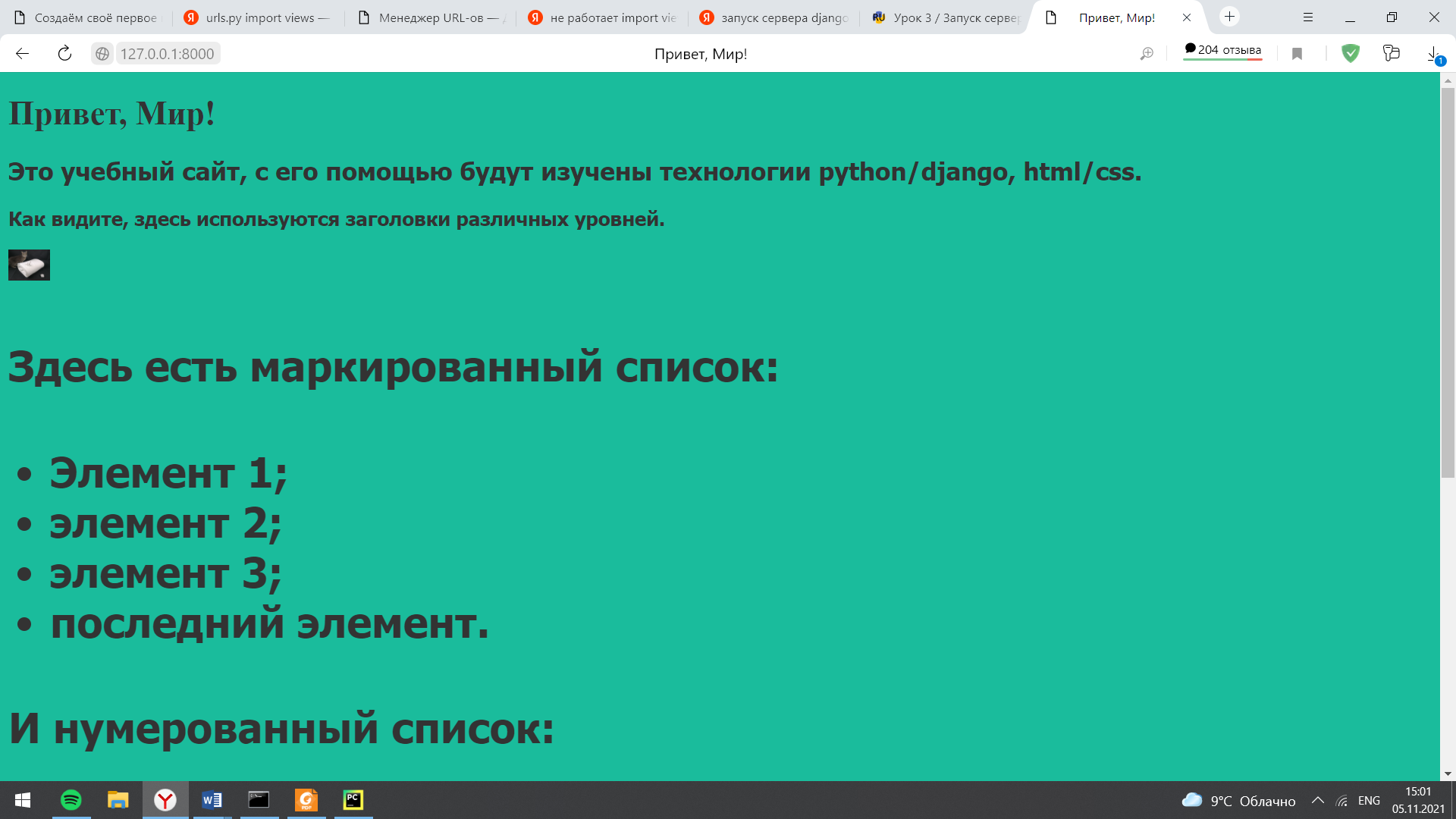


Рисунок 21 – Измененная картинка

Затем необходимо изменить размер шрифта для подзаголовка четвертого уровня. Для этого поменяем свойство «font-size:» с 20px на 40 px. Результат представлен на рисунке 22.

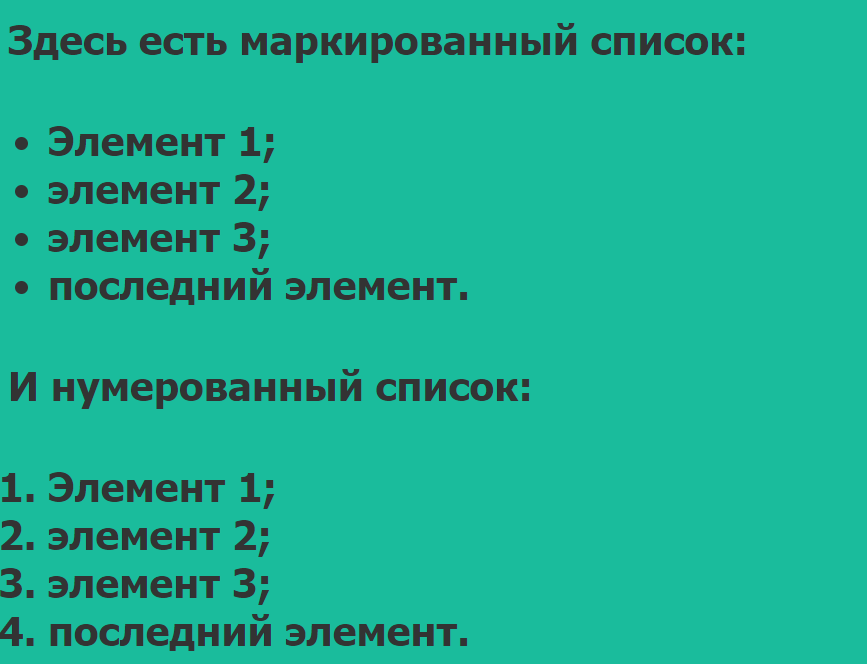


Рисунок 22 – Измененный заголовок четвертого уровня

Далее необходимо сделать таблицу на всю ширину экрана. Для этого в свойстве «width» выставим значение на 100%. Это представлено на рисунке 23.

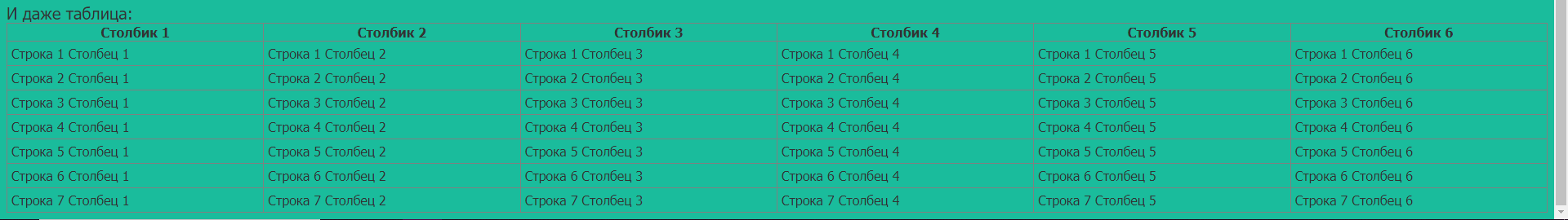


Рисунок 23 – Измененная таблица

Последним шагом лабораторной работы является загрузка проекта в удаленный репозиторий. Это представлено на рисунках 24-26.



Рисунок 24 – Загрузка проекта на удаленный репозиторий

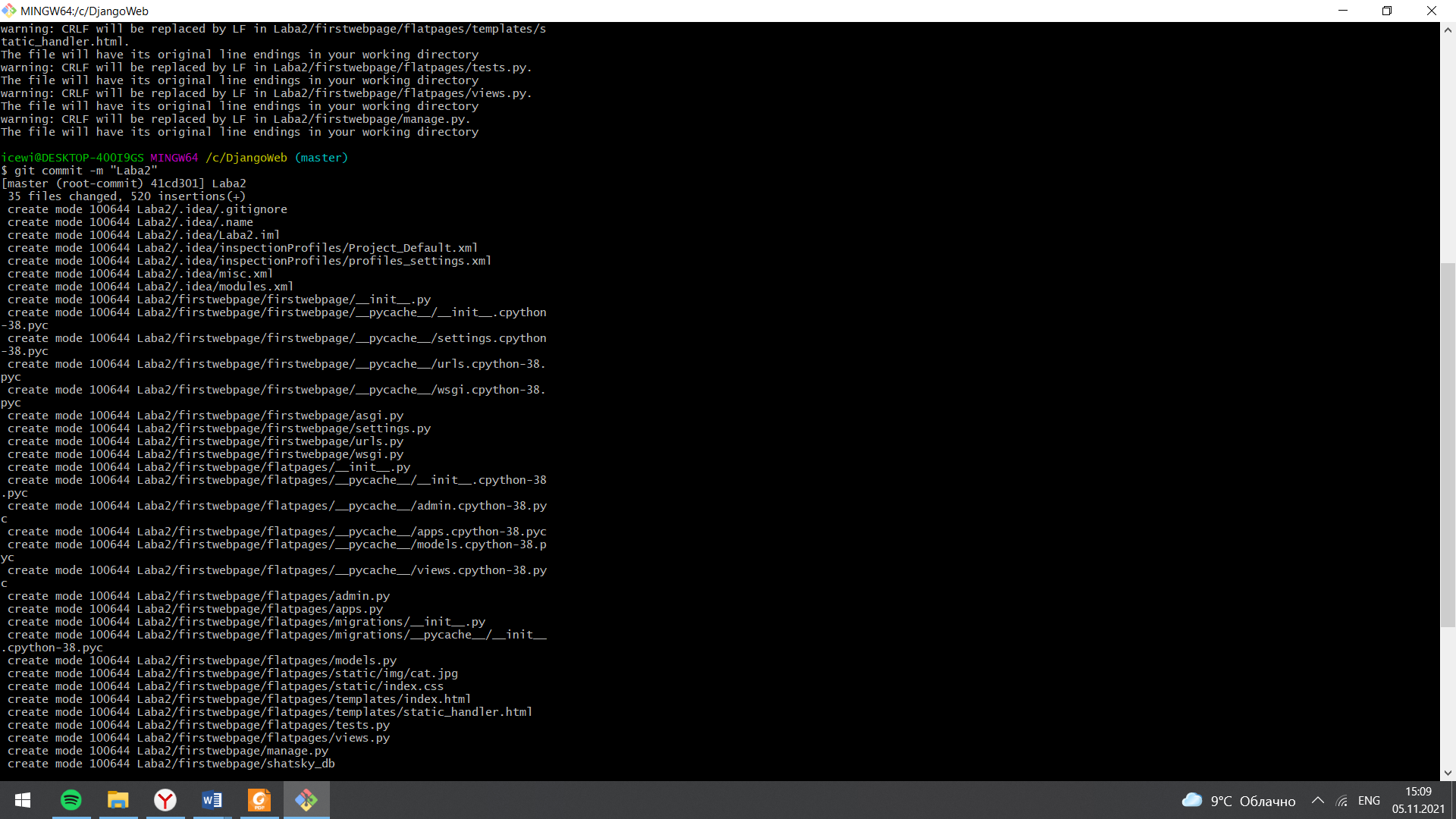


Рисунок 25 - Загрузка проекта на удаленный репозиторий(2)

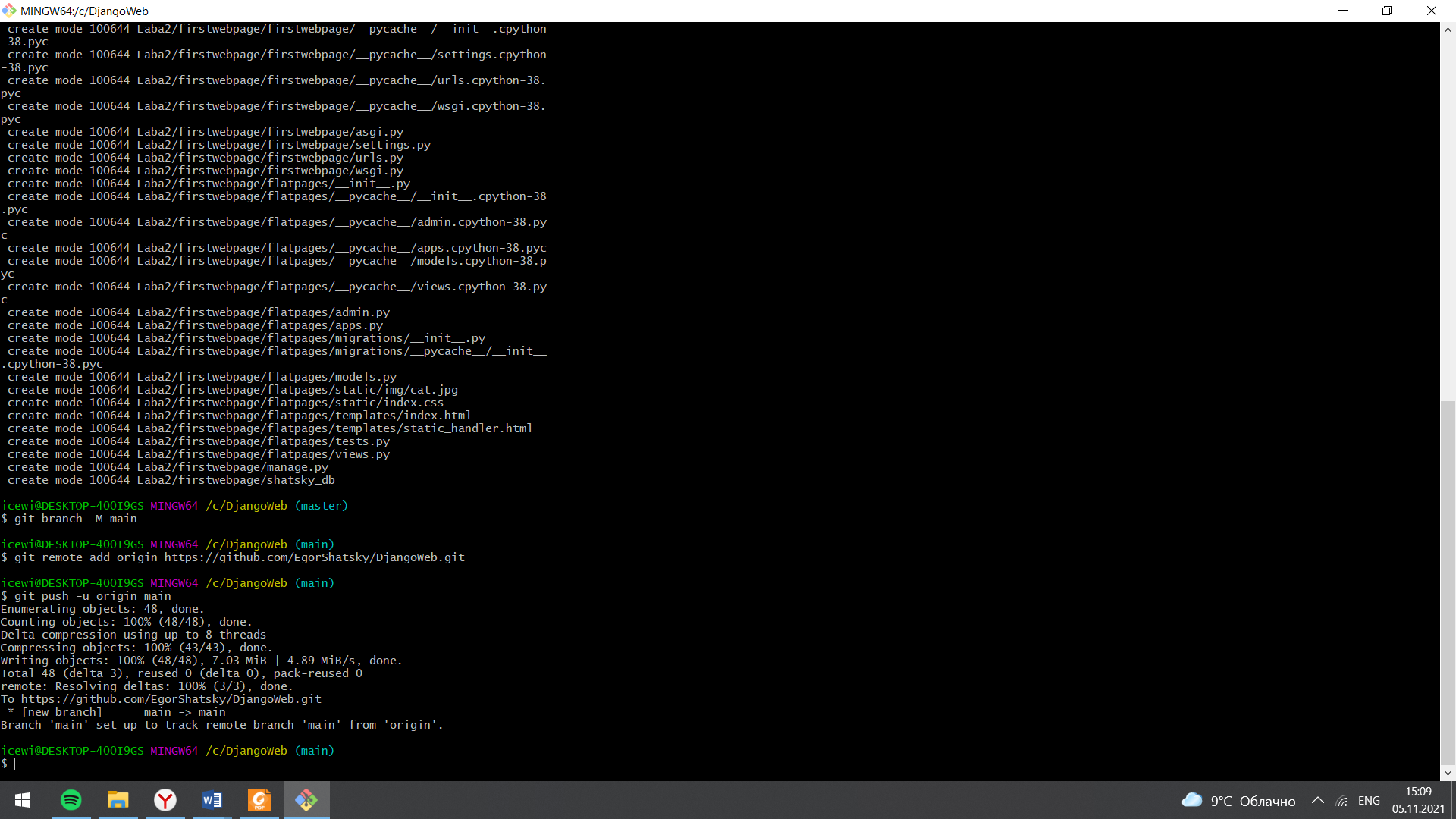


Рисунок 26 - Загрузка проекта на удаленный репозиторий(3)

**Вывод**: в этой лабораторной работы были изучены основы работы с шаблонами в Django и настройки для обработки статических фалов.

**Список используемых источников**

1 ГОСТ 7.32-2017 СИБИД. Отчет о научно-исследовательской работе. Структура и правила оформления.

2 ГОСТ 7.1-2003 Библиографическая запись. Библиографическое описание.