**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»**

Лабораторная работа №2

«Создание web-страницы с текстом»

по дисциплине:

«Web-программирование»

Выполнил:

Студент группы БФИ19

Бардюк Д.В.

Москва, 2021

**Задания:**

1. Добавьте к созданной таблице 2 строки и 2 столбца;
2. Добавьте границы для таблицы;
3. Сделайте заголовки списков (нумерованного и маркированного) подзаголовками четвертого уровня;
4. Создайте абсолютно идентичный шаблон, изменив только название на «static\_handler.html». В следующих заданиях при выполнении этой лабораторной работы изменяйте именно этот файл.
5. Добавьте к созданной таблице 2 строки и 2 столбца;
6. Добавьте границы для таблицы;
7. Сделайте заголовки списков (нумерованного и маркированного) подзаголовками четвертого уровня;
8. Создайте абсолютно идентичный шаблон, изменив только название на «static\_handler.html». В следующих заданиях при выполнении этой лабораторной работы изменяйте именно этот файл
9. Установите для заголовка первого уровня шрифт с засечками;
10. Добавьте картинку и сделайте ее высотой 30px;
11. Измените размер шрифта для подзаголовков четвертого уровня;
12. Сделайте ширину таблицы на 100% экрана;
13. Загрузите ваш проект на любой гит-репозиторий (GitHub, GitLab, Google Code, Bitbucket и т.п.)

**2. Ход работы**

Создаём новую директорию для лабораторной работы №2 и создаём новый проект firstwebpage, подключая приложение flatpage, выполнив команды как показано на рисунке 1.

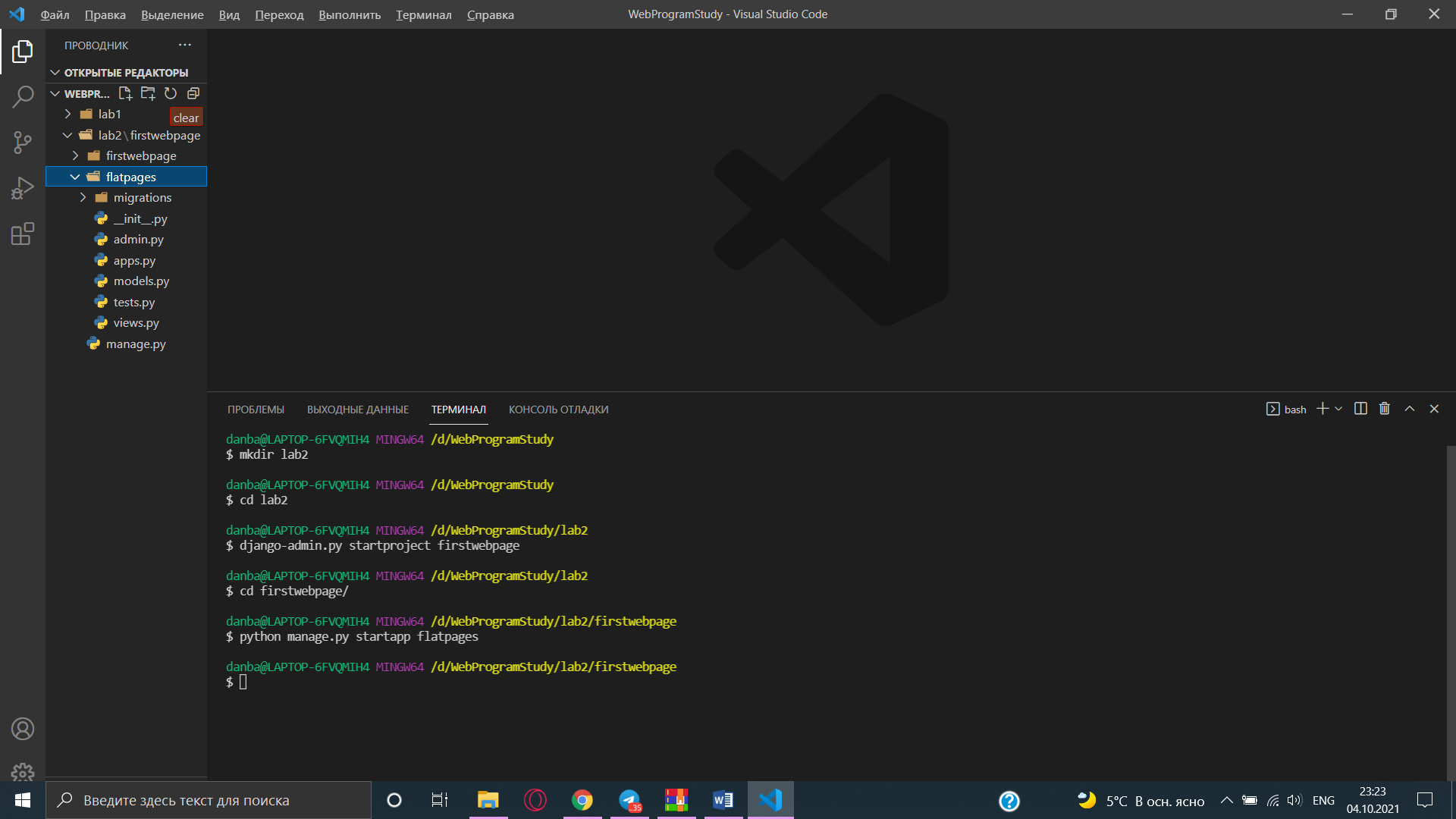


Рисунок 1 – Создание нового проекта

В начале работы над новым проектом задаём базовые настройки: имя базы данных и создаём ее таблицы, а также добавим нужные приложения в проект, выполнив команды как на рисунке 2. Чтобы добавить приложение flatpages в свой проект, в файле settings.py, в кортеже INSTALLED\_APPS добавил в конец элемента строку 'flatpages', как показано на рисунке 3.

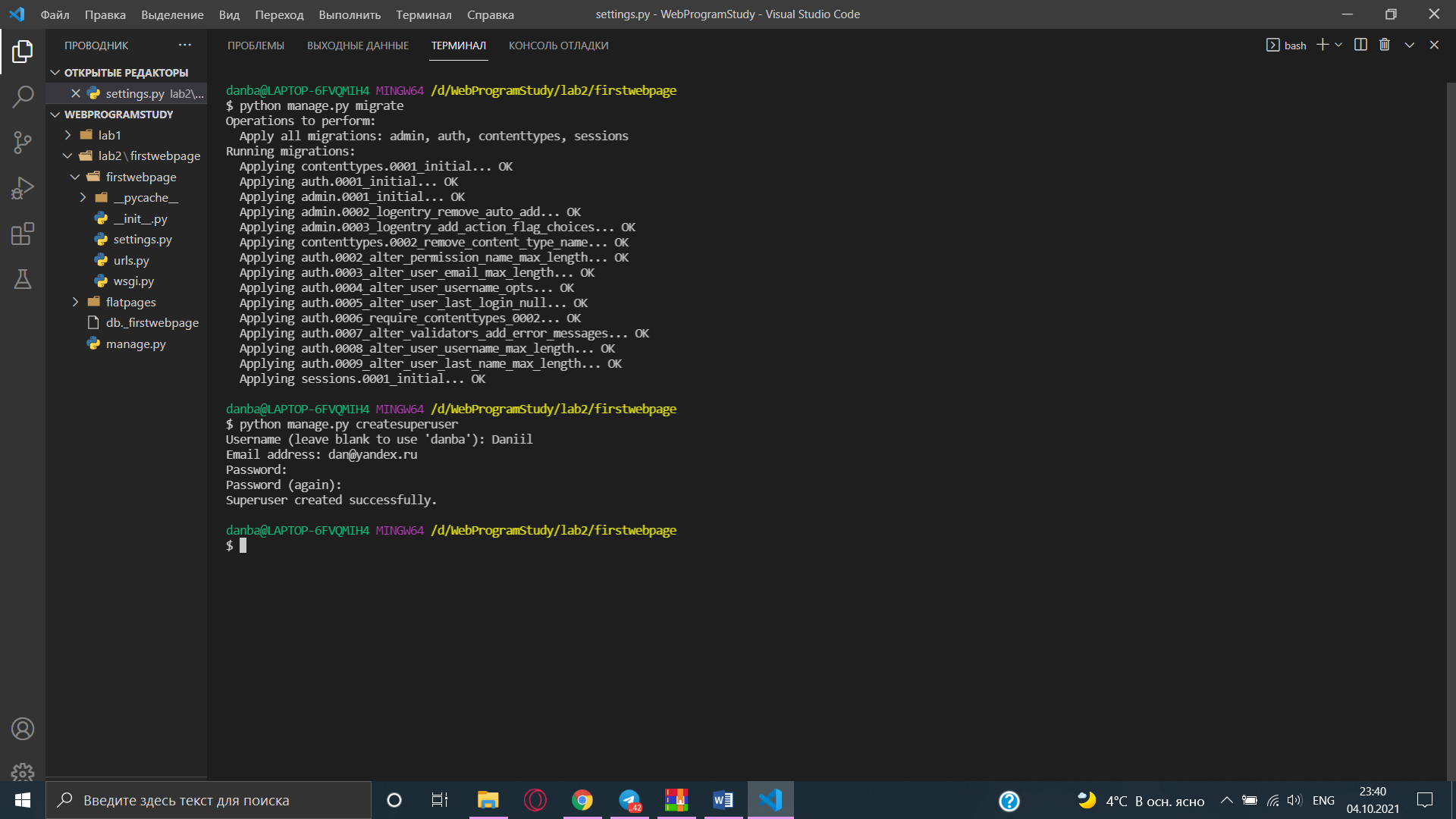


Рисунок 2 – Базовые настройки проекта

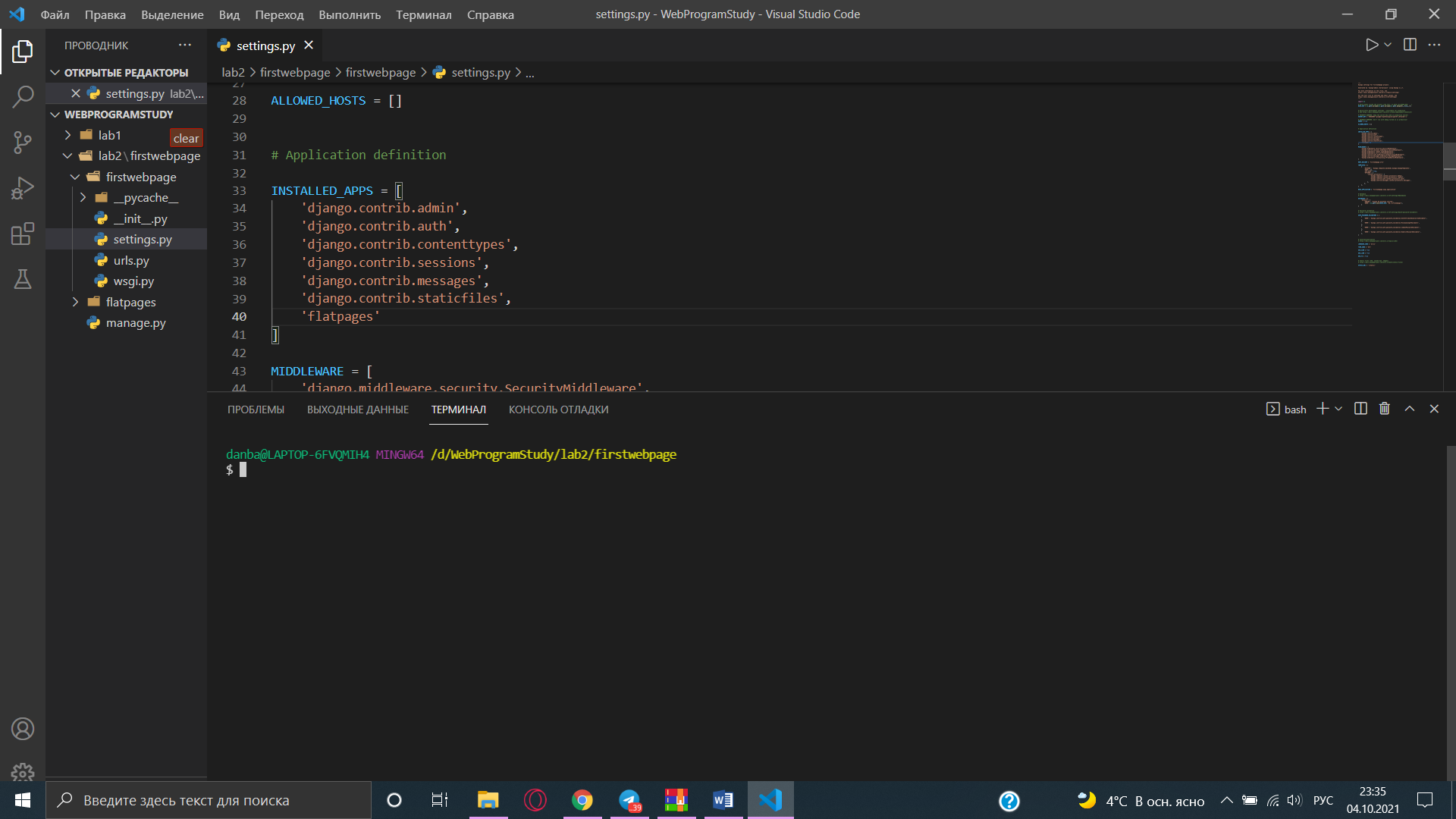


Рисунок 3 – Подключение flatpages

Перед началом работы с файлом urls.py необходимо импортировать views нашего приложения и для будущей страницы создать новый адрес в файле urls.py, как показано на рисунке 4.

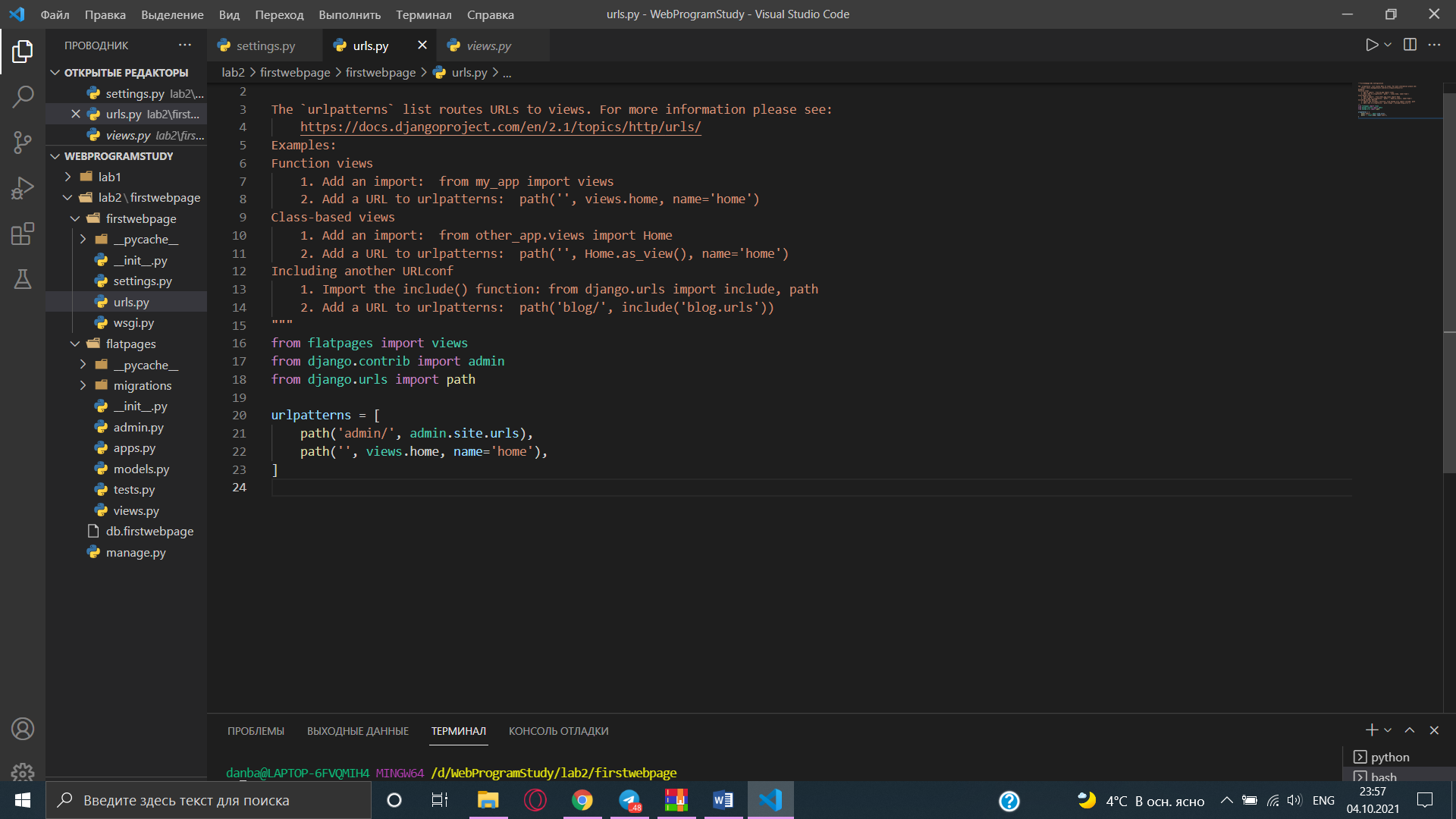


Рисунок 4 – Работа с файлом urls.py

Второй аргумент, передаваемы 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, создадим функцию home в файле views.py в директории flatpages, как показано на рисунке 5.

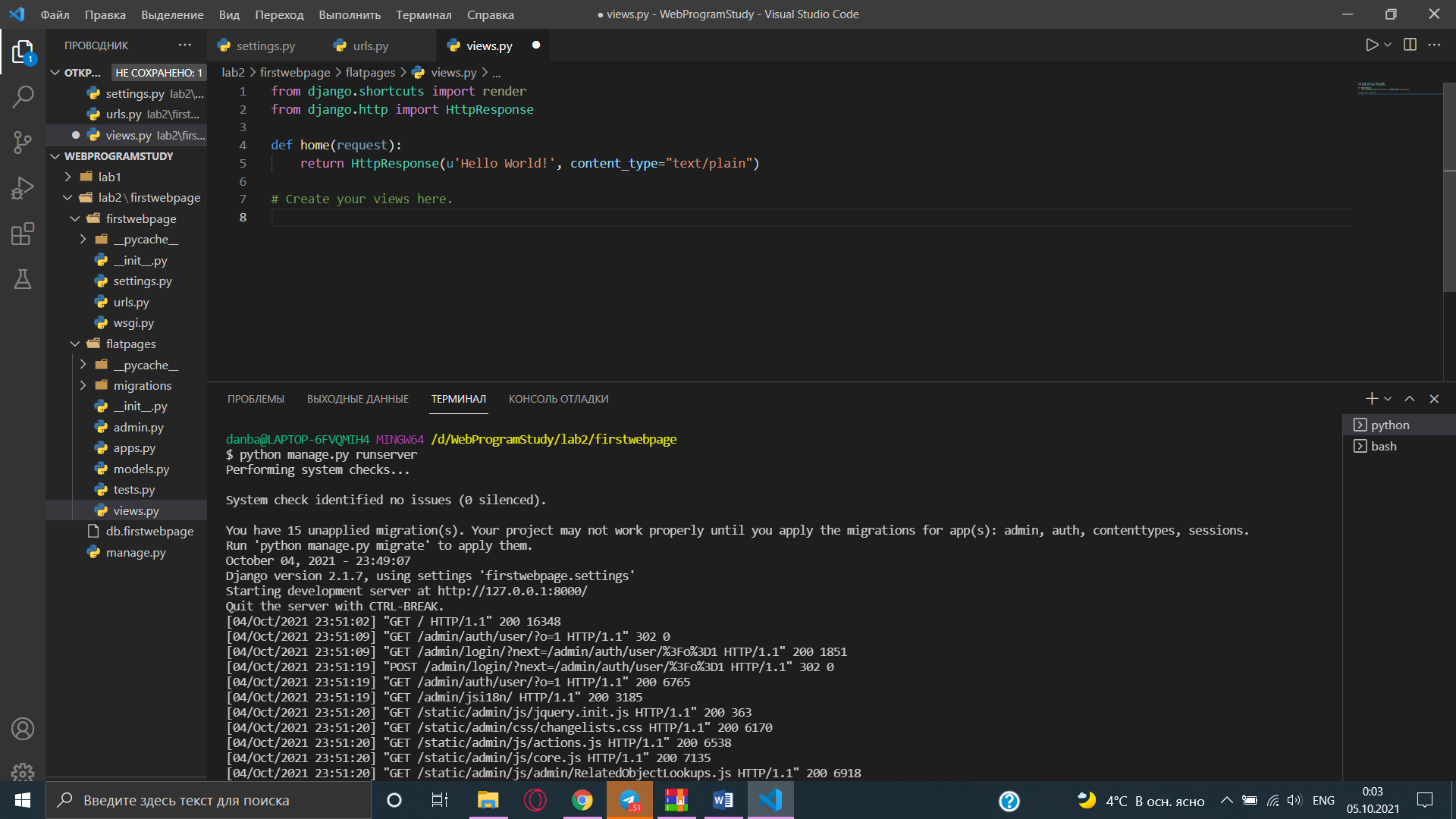


Рисунок 5 – Работа с файлом views.py

При переходе на страницу мы видим результат работы этой функции. Рисунок 6.

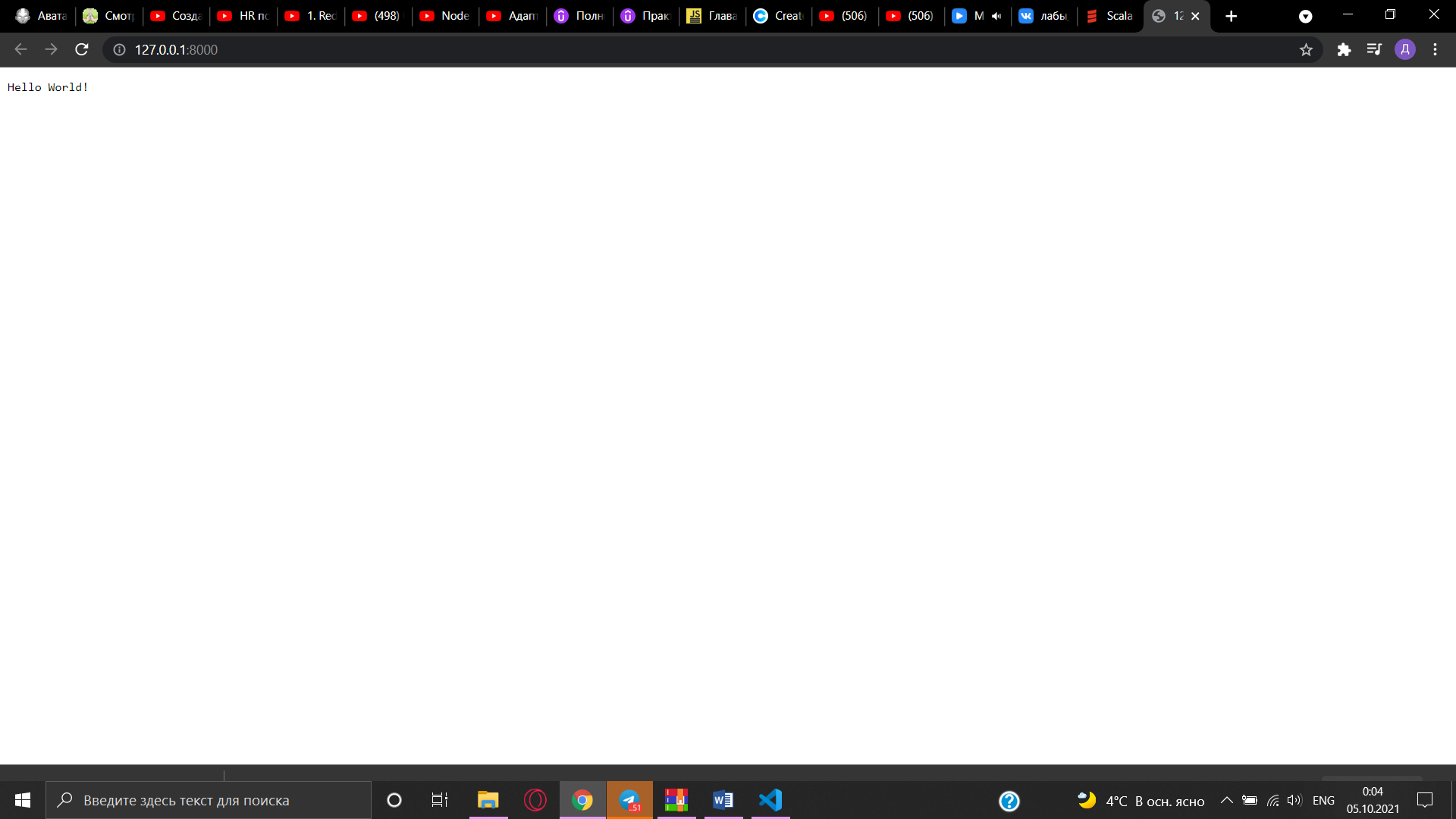


Рисунок 6 – Начальная страница

Чтобы по адресу http://127.0.0.1:8000/hello/ возвращался тот же самый текст нужно в файле urls.py сделать как показано на рисунке 7. Результат представлен на рисунке 8.

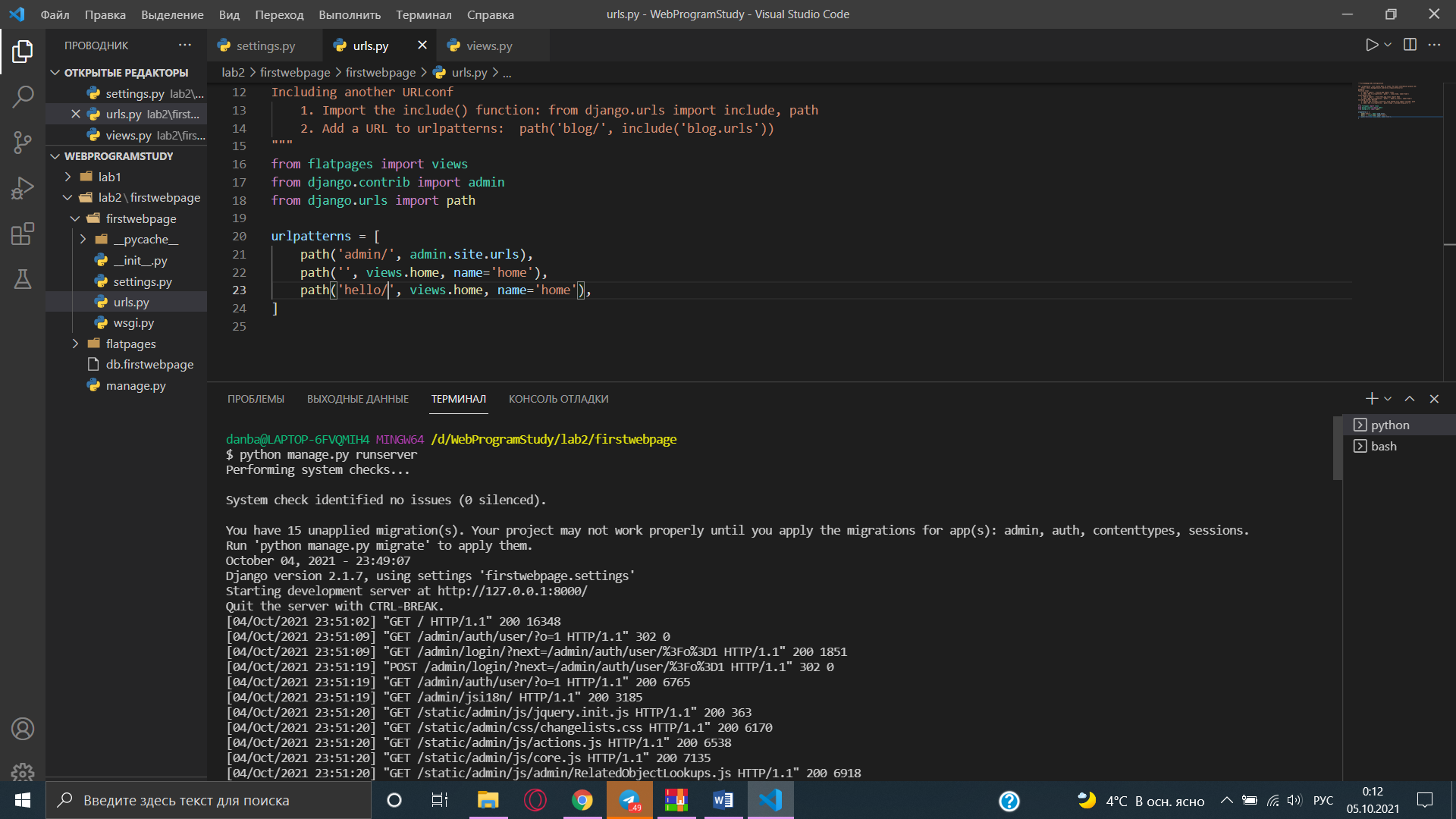


Рисунок 7 – Добавление пути /hello

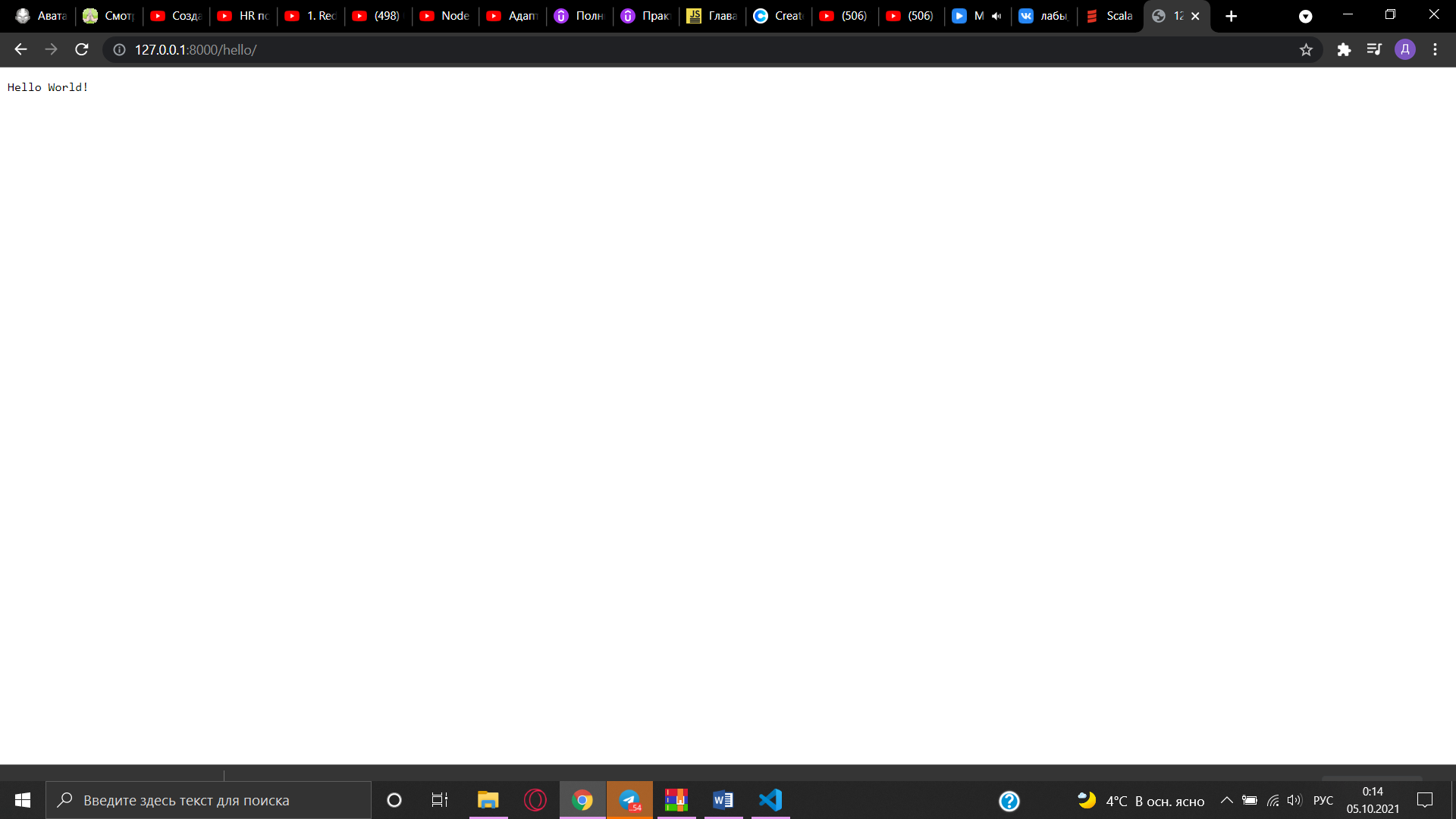


Рисунок 8 – Вывод текста на странице /hello

Уберём тип указание типа возвращаемого ответа у функции home (рисунок 9). Результат представлен на рисунке 10. Можно заметить, что у нашего текста изменился шрифт.

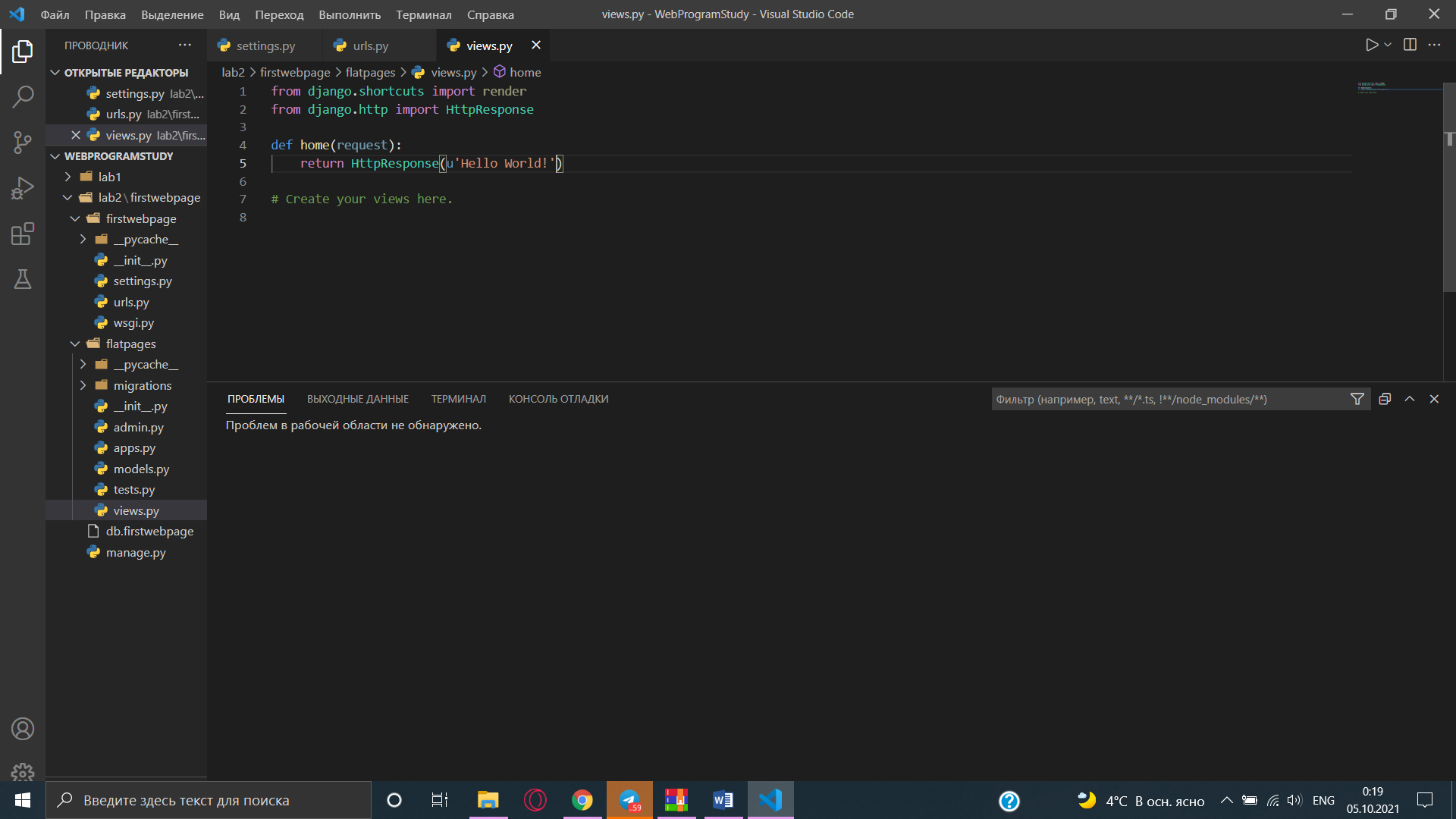


Рисунок 9 – Функция home

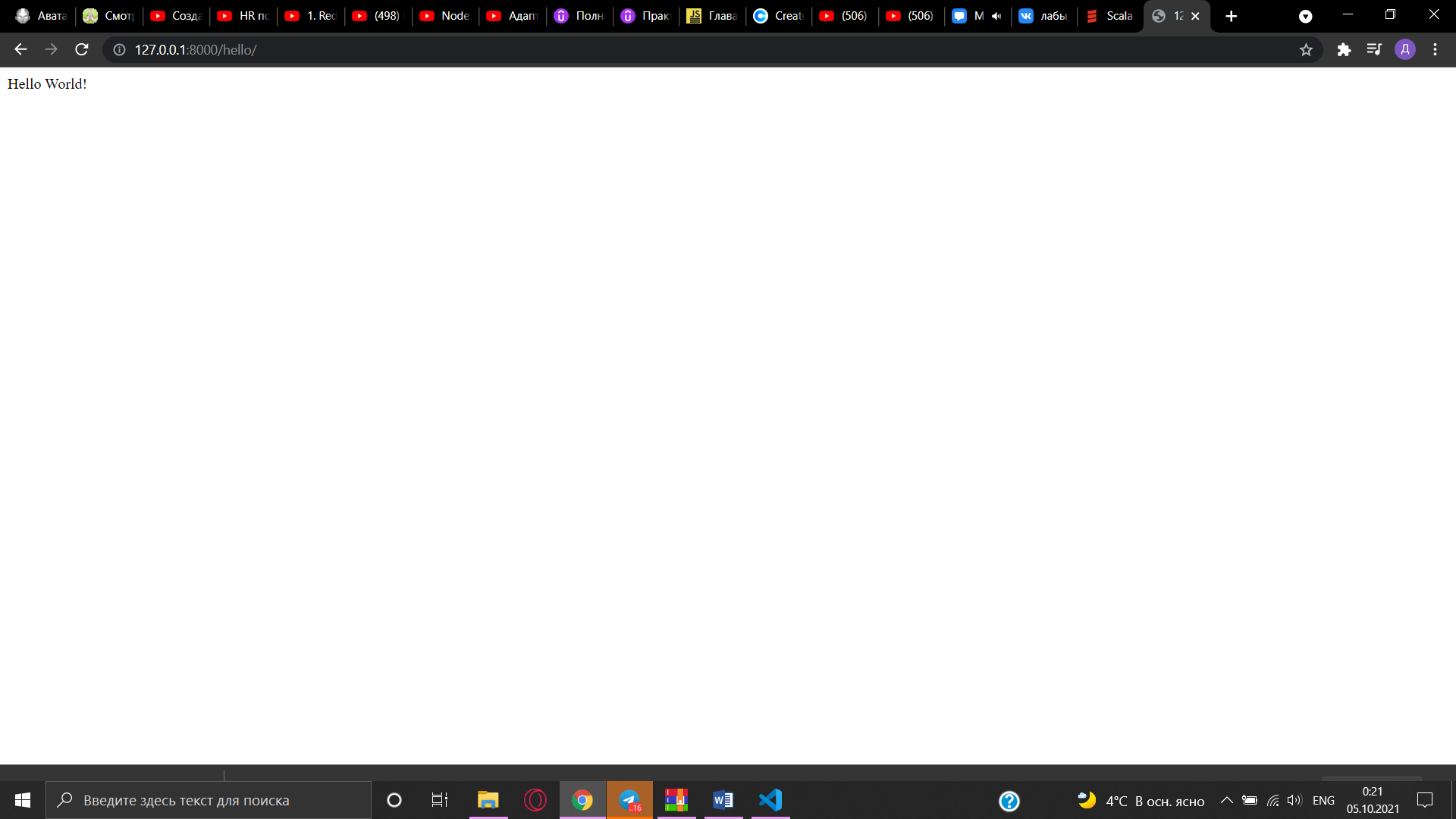


Рисунок 10 – Результат работы

Создайте папку templates в директории flatpages. Затем, в папке templates создайте файл index.html как показано на рисунке 11.

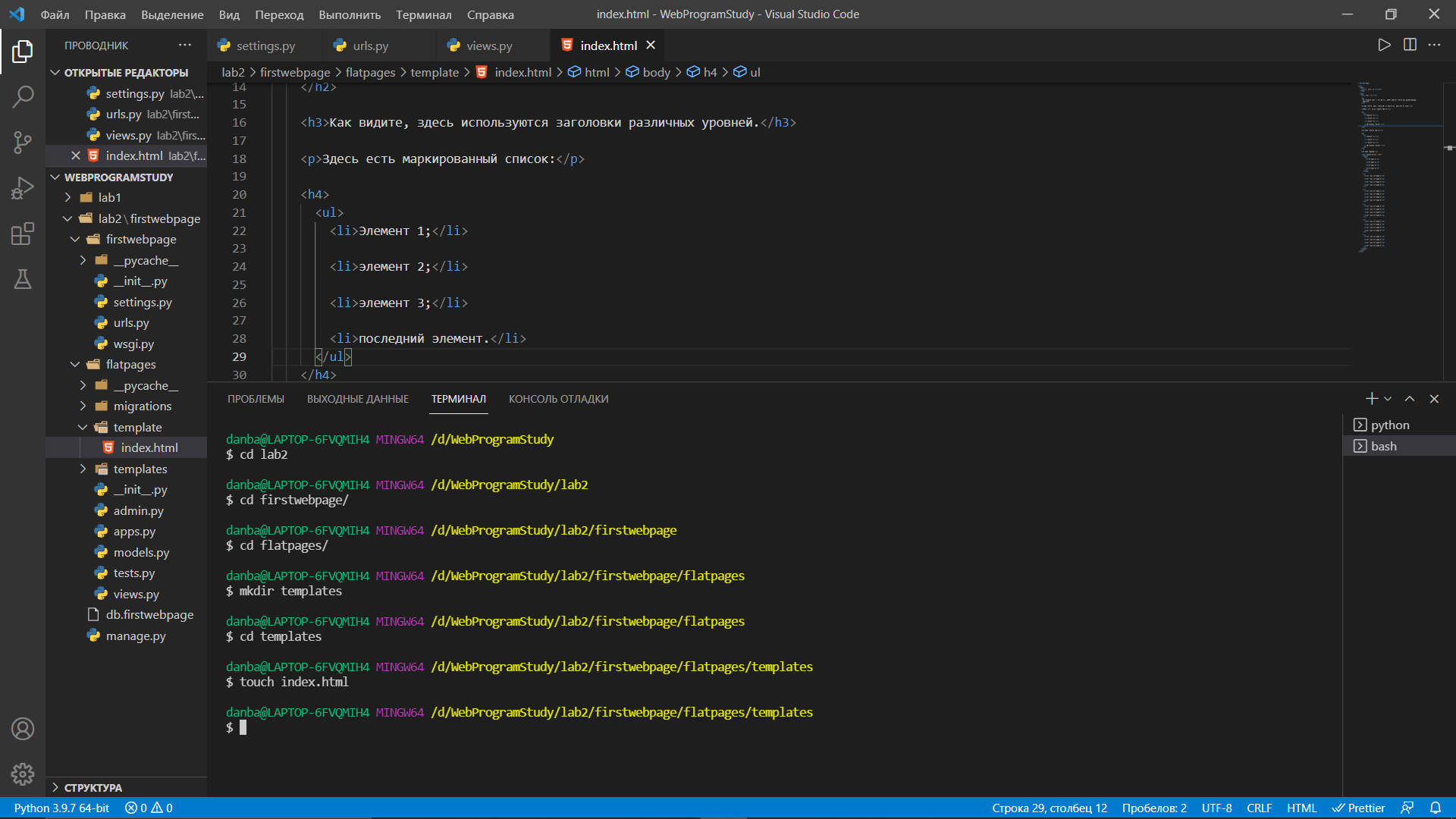


Рисунок 11 – Создание html документа

Код index.html представлен в листинге 1.

**Листинг 1**

<!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>

Чтобы вместо простого текстового ответа приходил html-документ, созданный html-файл необходимо подключить к функции-представлению home. Для этого в файле views.py, который находится в директории flatpages, добавьте операции импортирования и функцию-представление home изменим как на рисунке 12

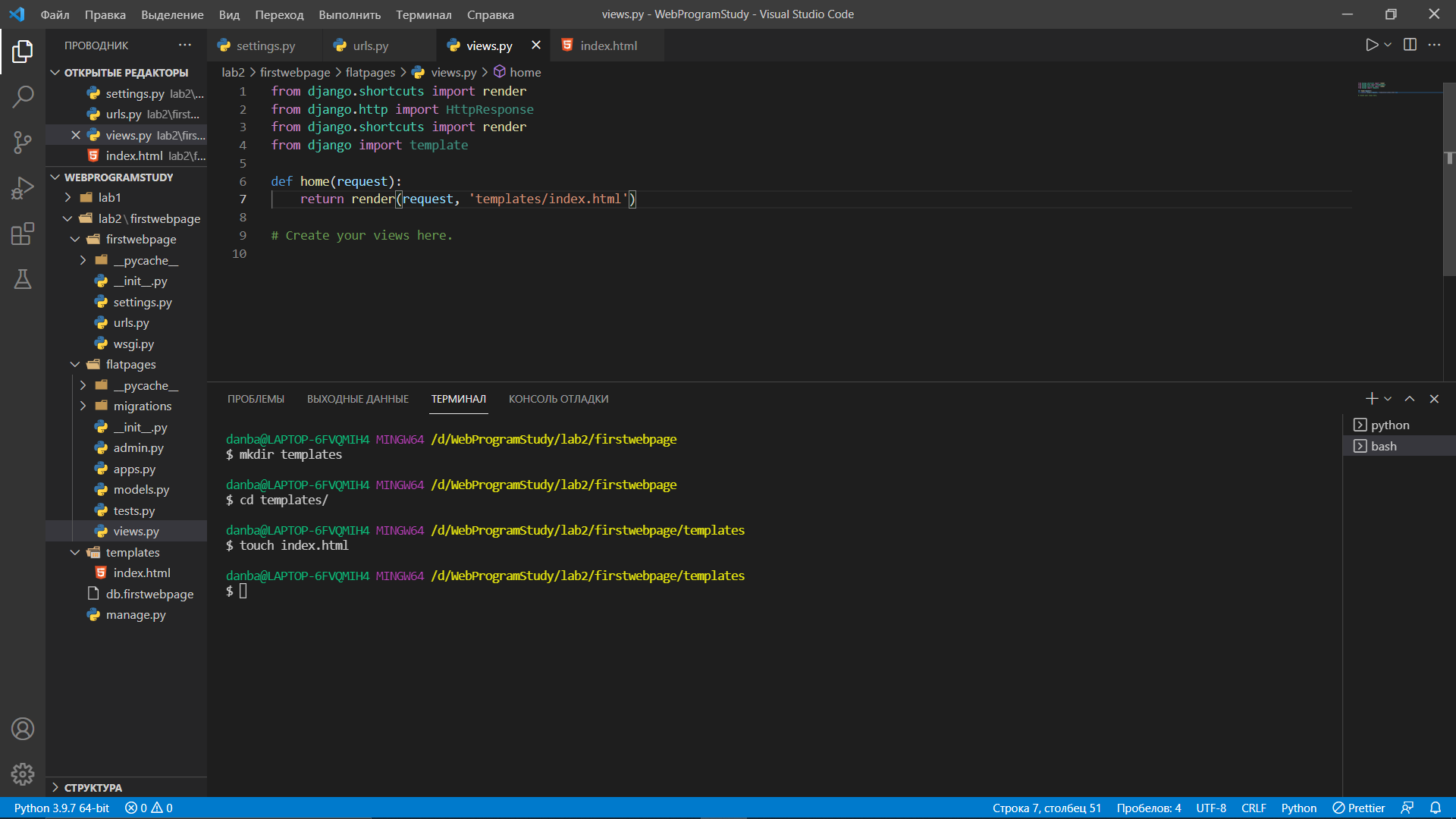


Рисунок 12 – Файл views.py

Также, для того, чтобы файл index.html был найден в директории templates, в файле settings.py изменим поле DIRS кортеже TEMPLATES. Поле DIRS должно содержать адрес директории, в которой располагается файл index.html как на рисунке 13.

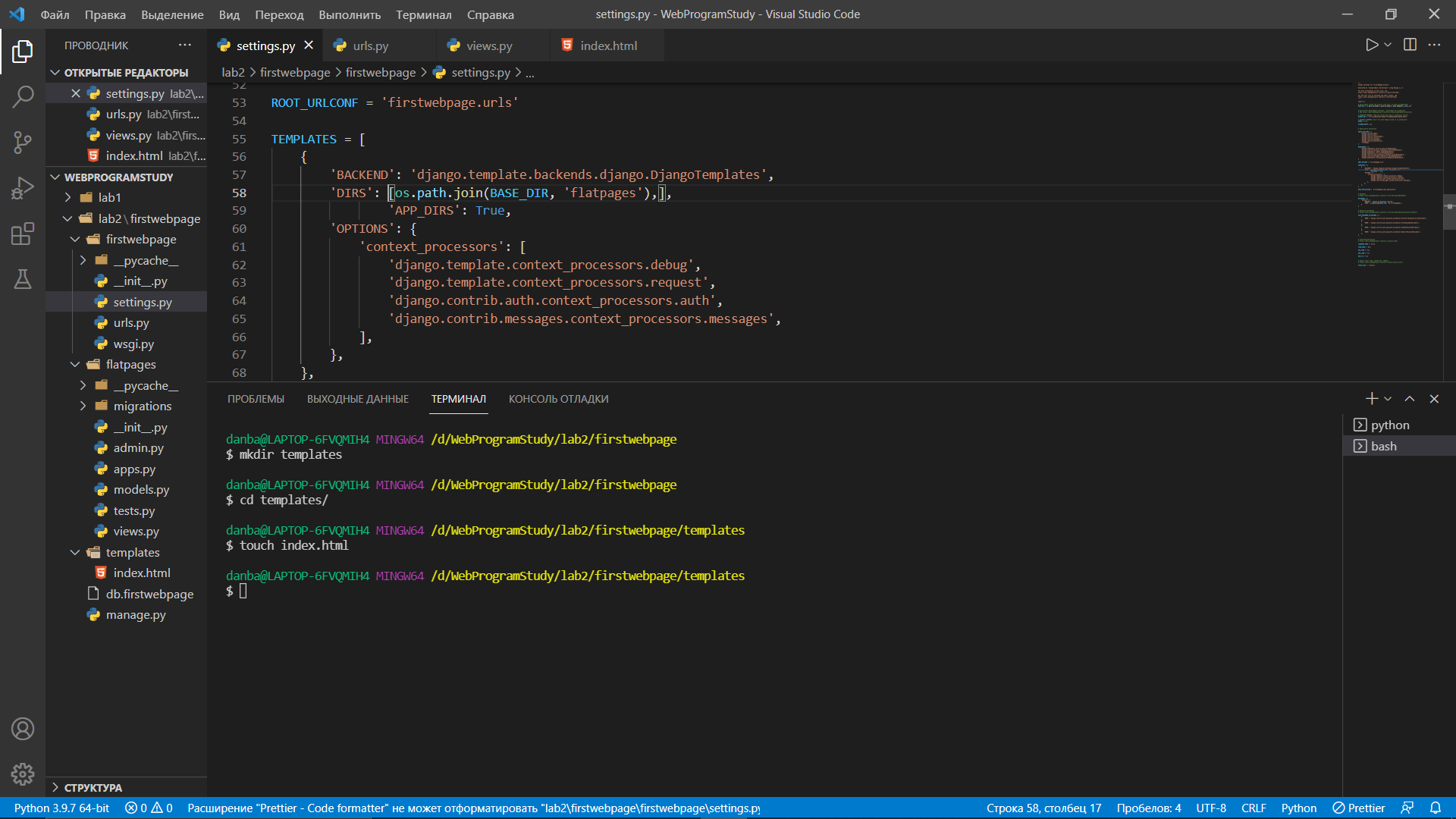


Рисунок 13 – Изменение поля DIRS

Результат работы нашей страницы представлен на рисунке 14.

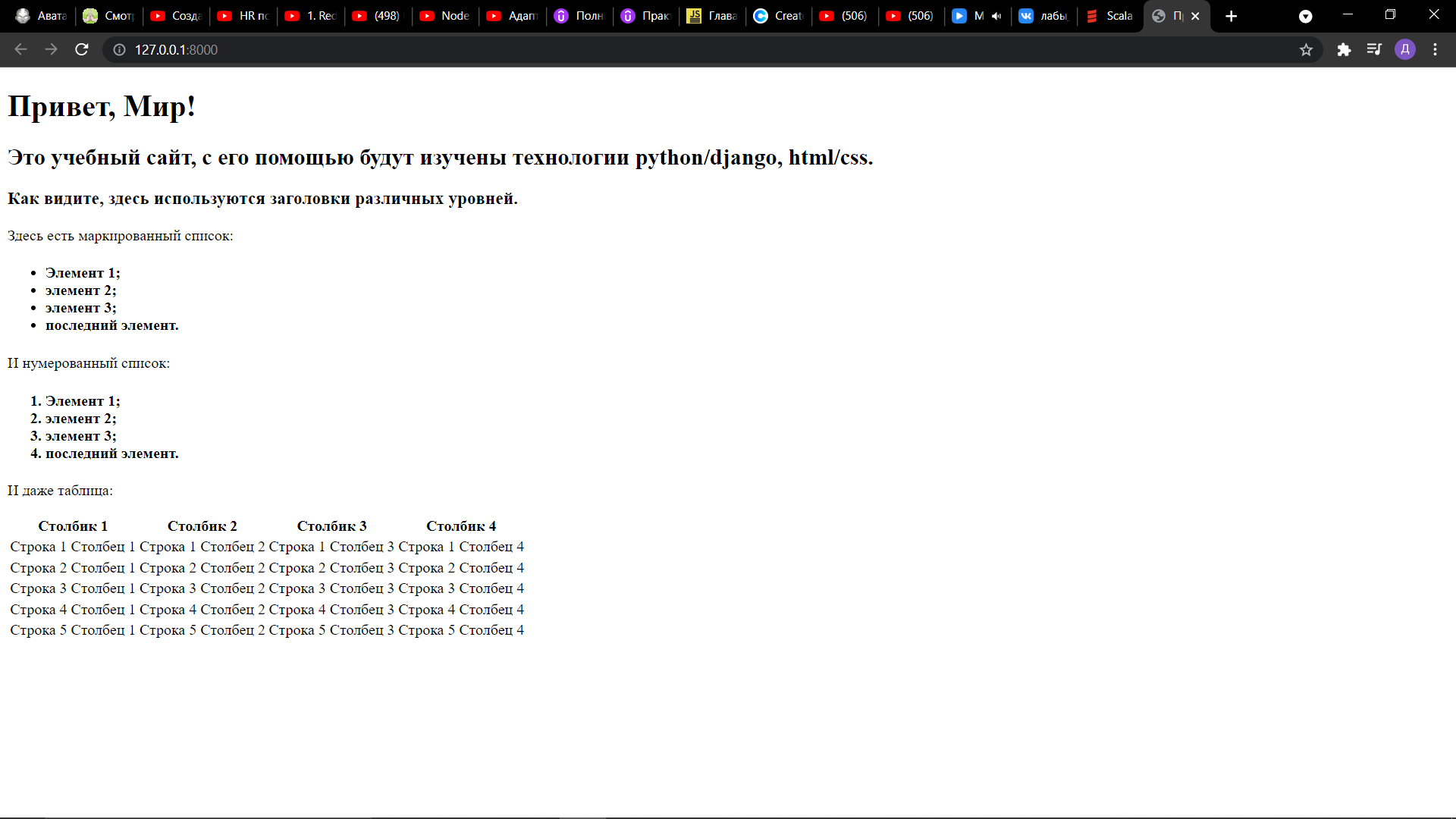


Рисунок 14 – Основная страница

Создадим ещё 2 столбца и 2 строки к заданной таблице. Результат представлен на рисунке 15.

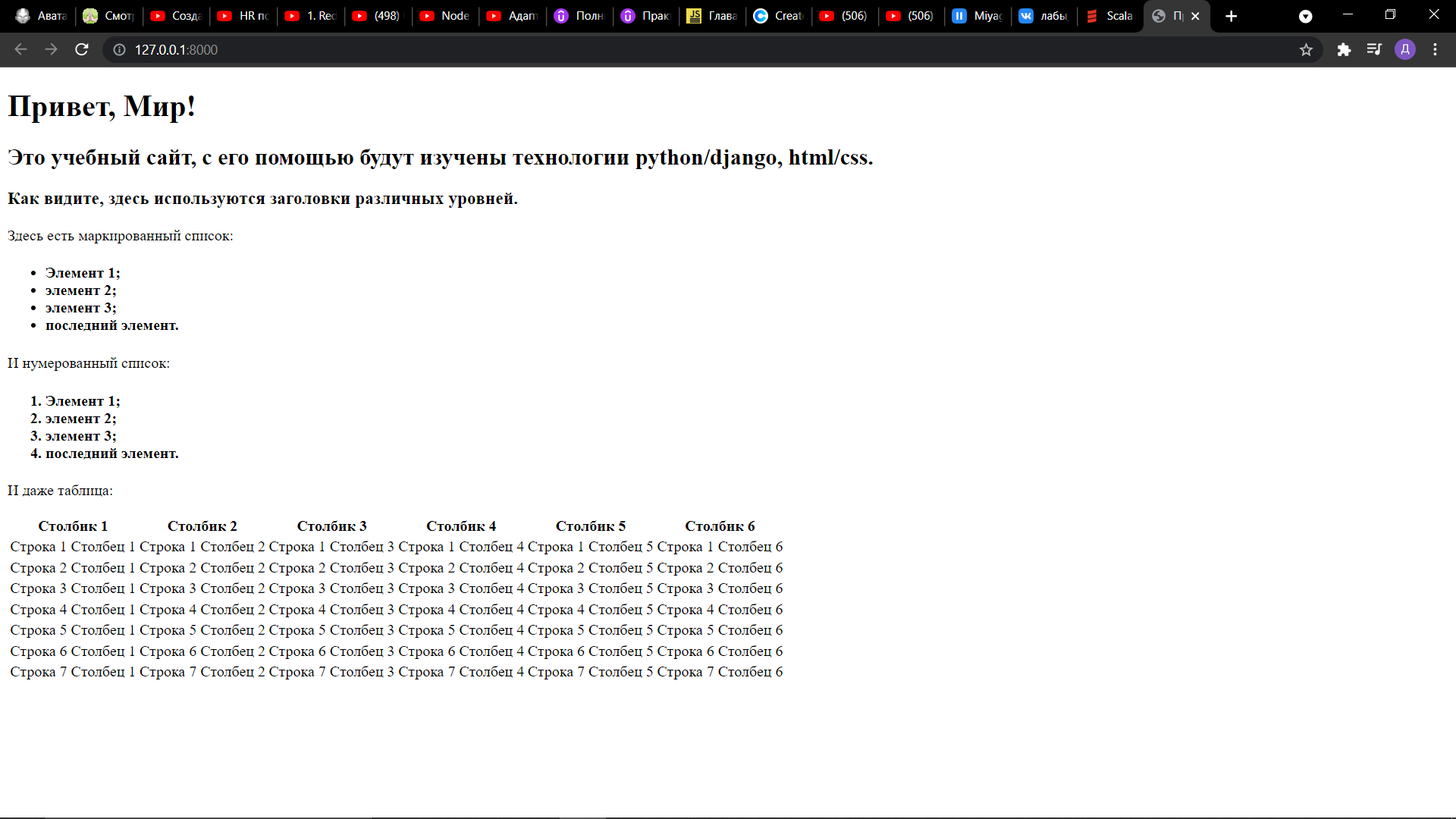


Рисунок 15 – Добавление столбцов и строк

Добавим границы нашей таблице. Результат на рисунке 16.

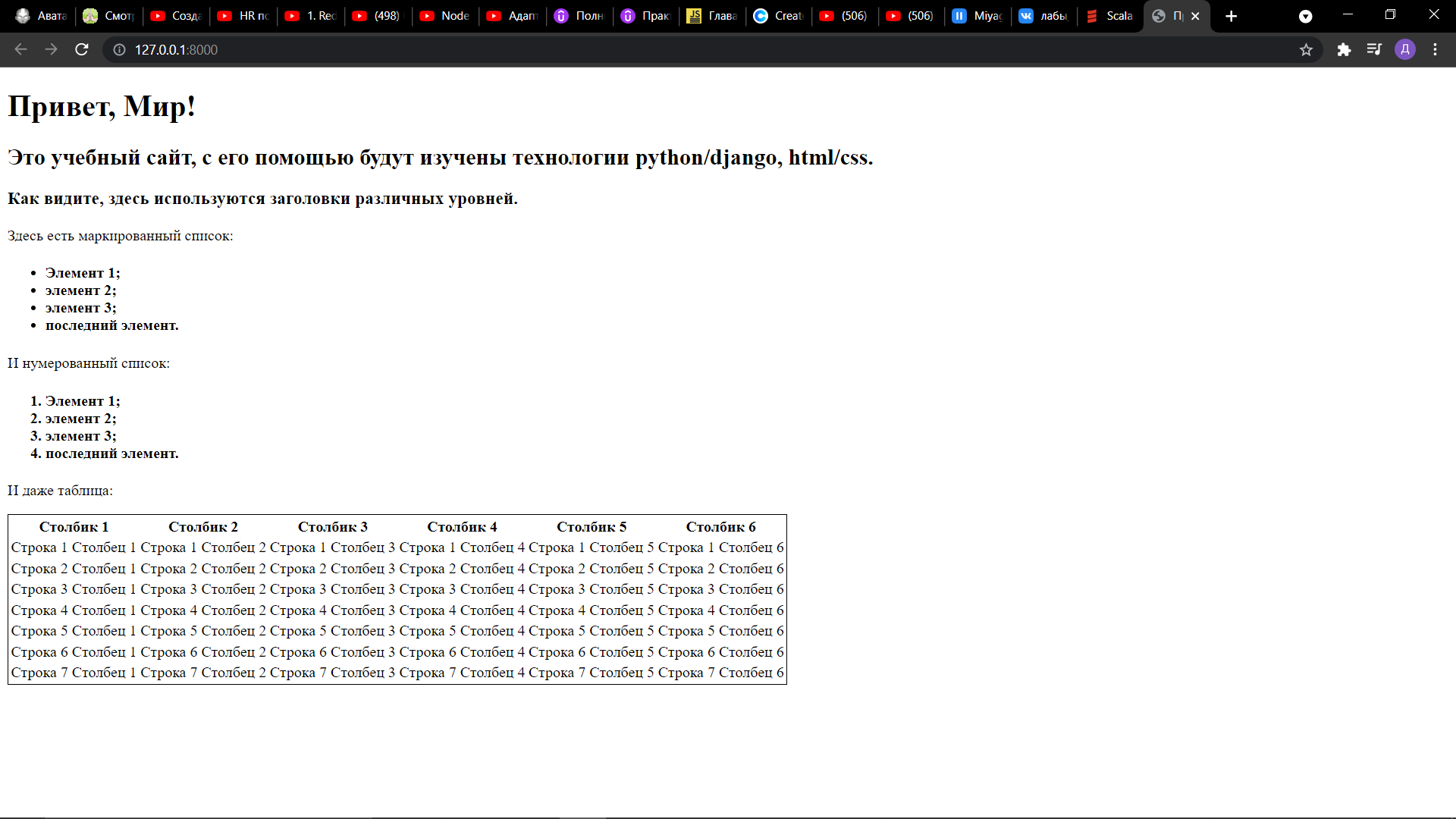


Рисунок 16 – Таблица с границей

Сделаем заголовки списков подзаголовками 4 уровня. Результат представлен на рисунке 17.

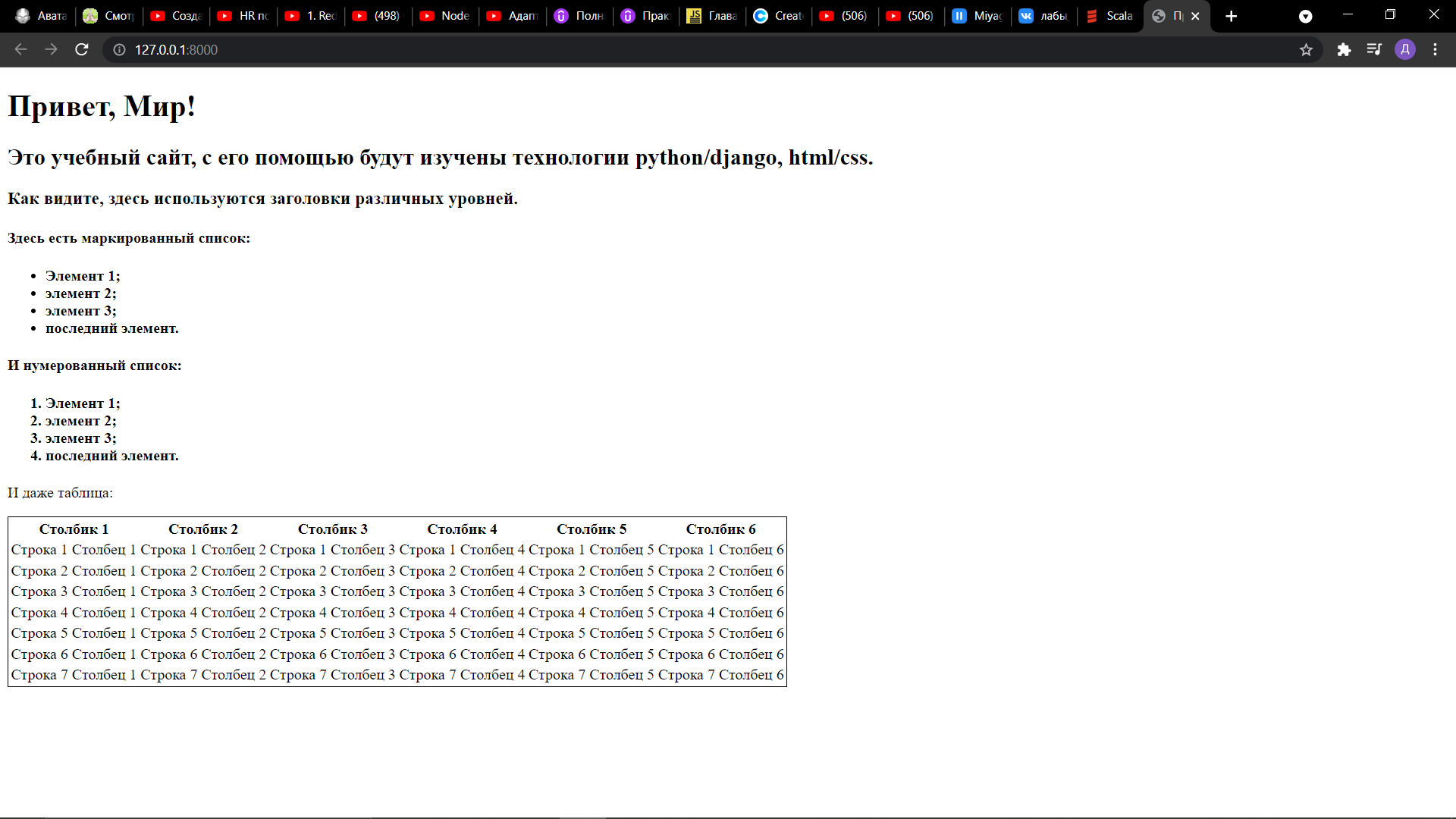


Рисунок 17 – Измененные заголовки.

Создадим идентичный шаблон под названием «static\_handler.html». Результат представлен на рисунке 18.

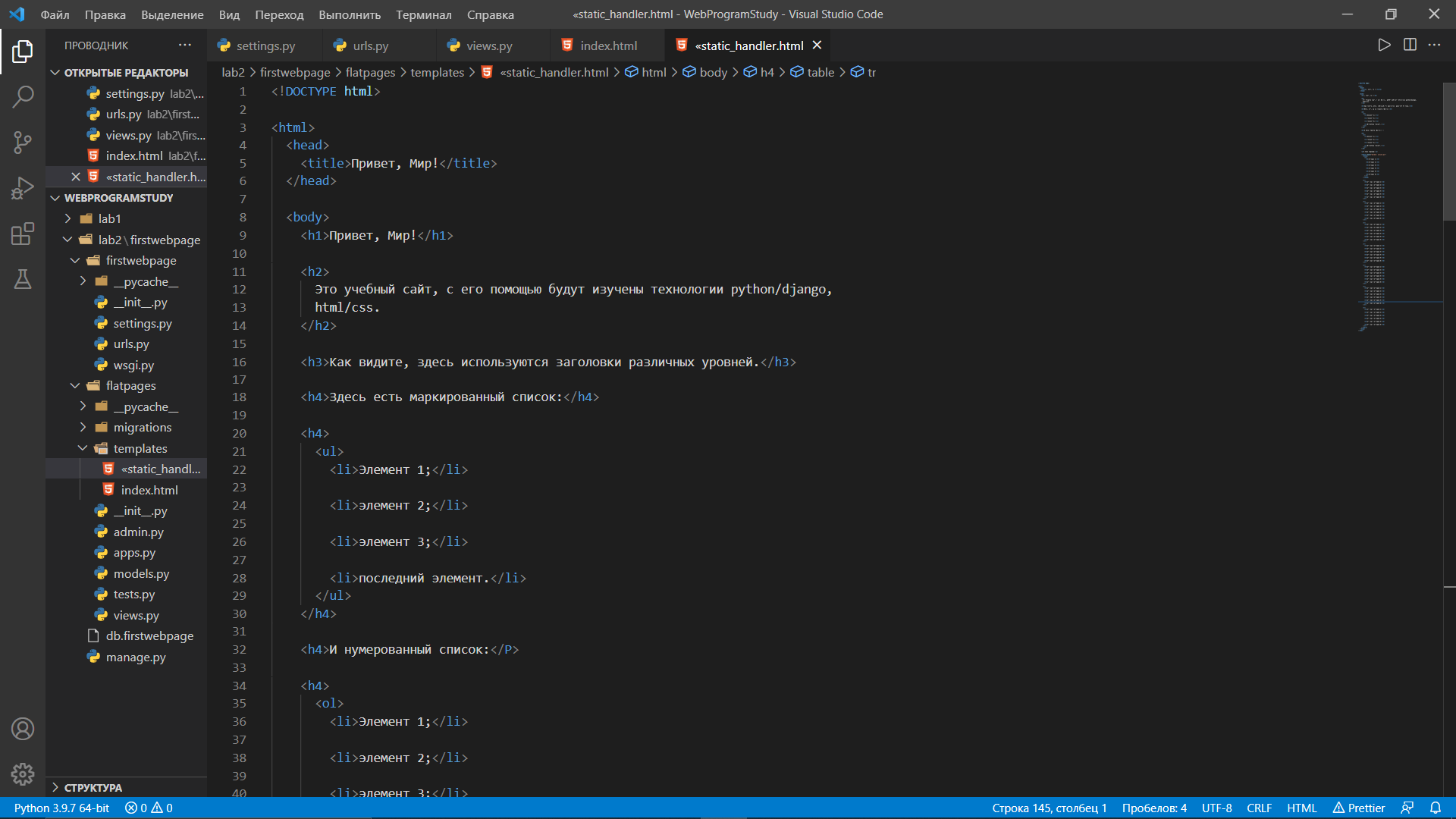


Рисунок 18 – Шаблон

Для того, чтобы придать документу стиль в соответствии с макетом, будем использовать CSS (формальный язык описания внешнего вида документа, написанного с использованием языка разметки). Создим папку static в директории flatpages, а в ней файл index.css как показано на рисунке 19.

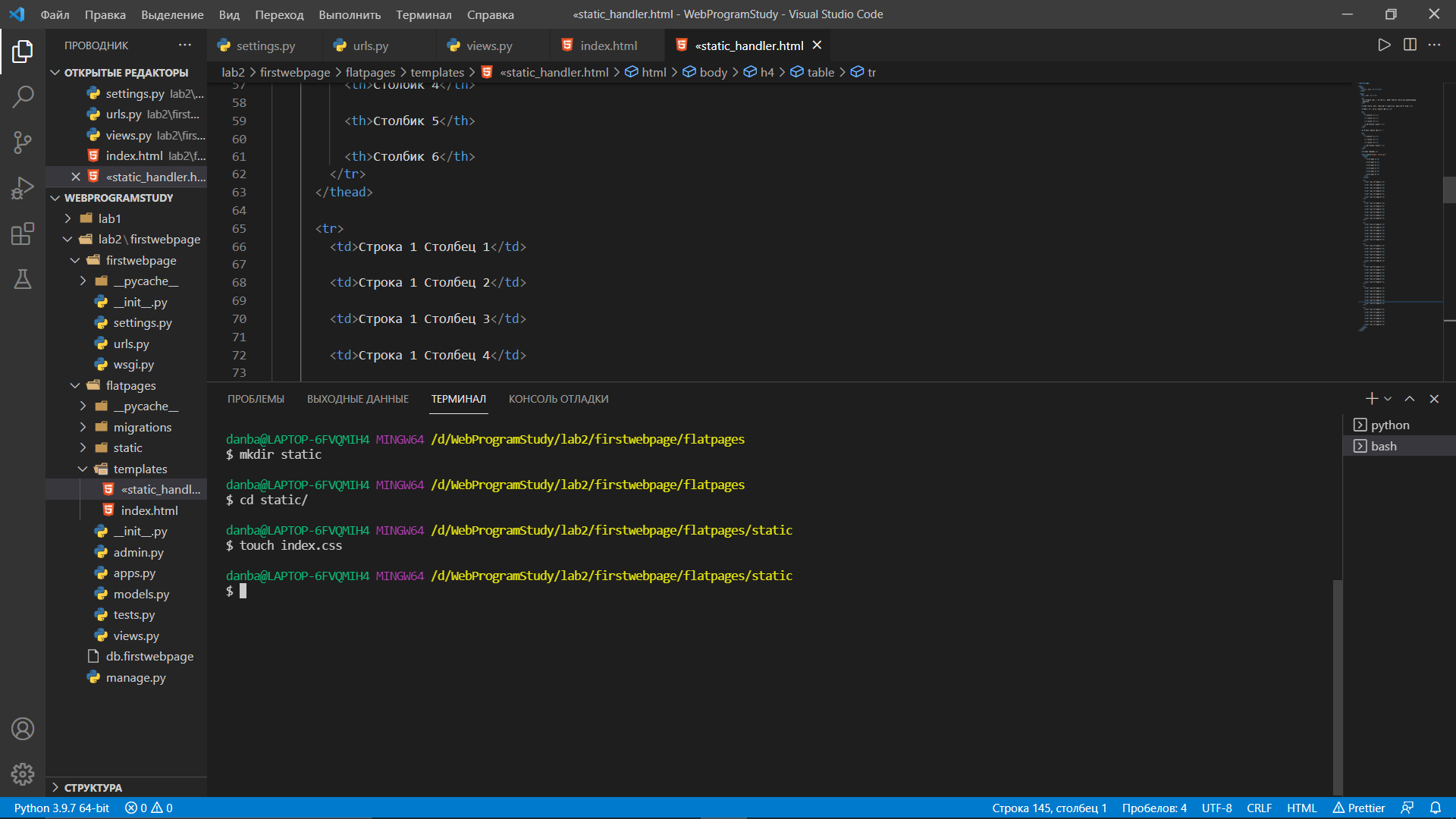


Рисунок 19 – Создание файла index.css

Код файла index.css представлен в листинге 2.

**Листинг 2**

body {

background: #1abc9c;

font-family: Tahoma, Arial, sans-serif;

color: #333;

}

table {

border-collapse: collapse;

}

p, h4 {

font-size: 20px;

margin-bottom: 0;

}

h4 {

font-size: 14px;

}

ul, ol {

margin: 0;

}

table tr td {

padding: 5px;

}

table {

width: 100%;

}

img {

height: 30px;

width: auto;

}

В файл страницы static\_handler.html во внутрь тега <head> вставьте тег подключения css-скрипта. Результат представлен на рисунке 20.

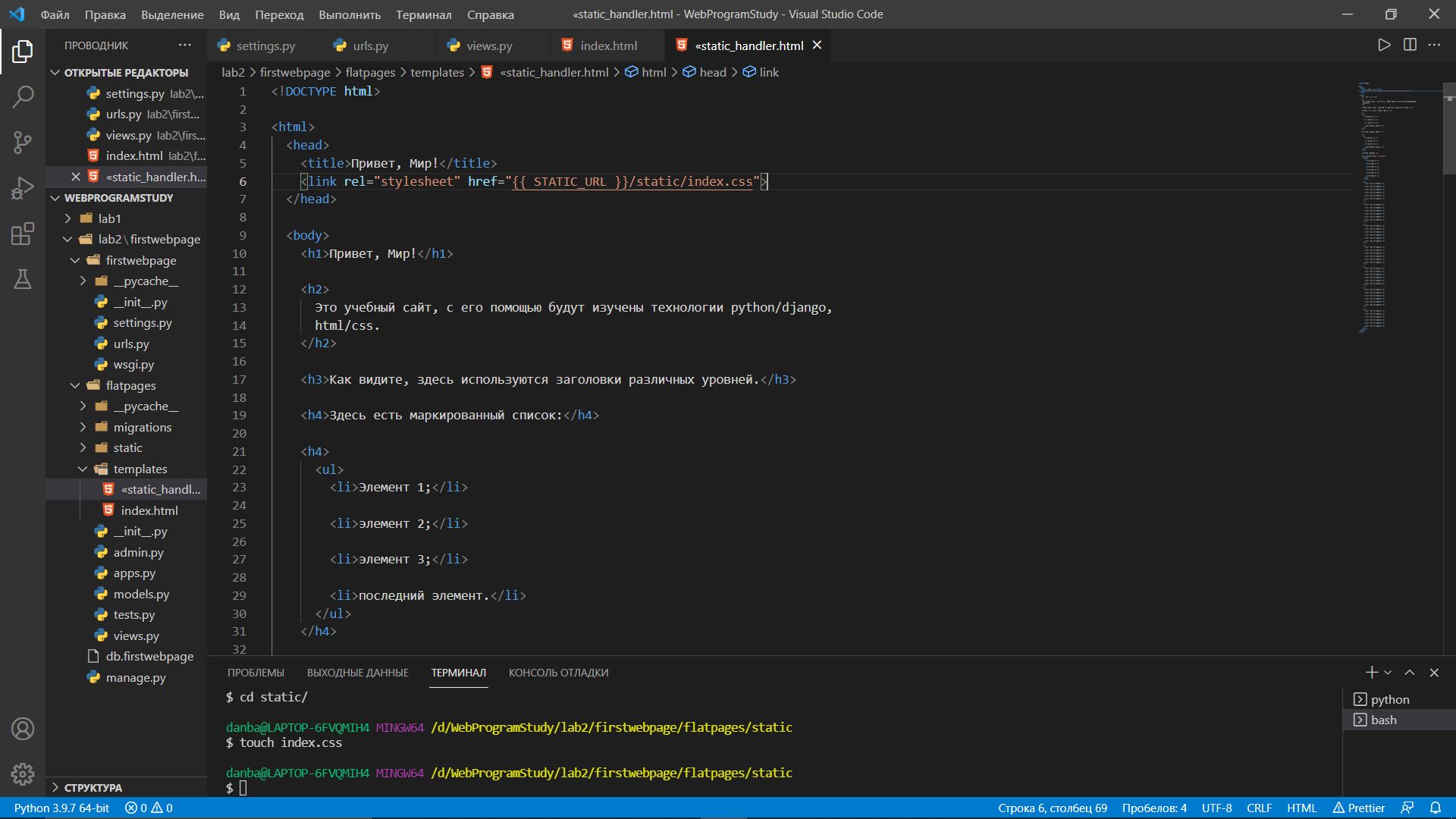


Рисунок 20 – Подключение css файла

Установим для заголовка первого уровня шрифт с засечками, добавим картинку и сделайте ее высотой 30px, изменим размер шрифта для подзаголовков четвертого уровня, сделаем ширину таблицы на 100% экрана. Код представлен на рисунке 21.

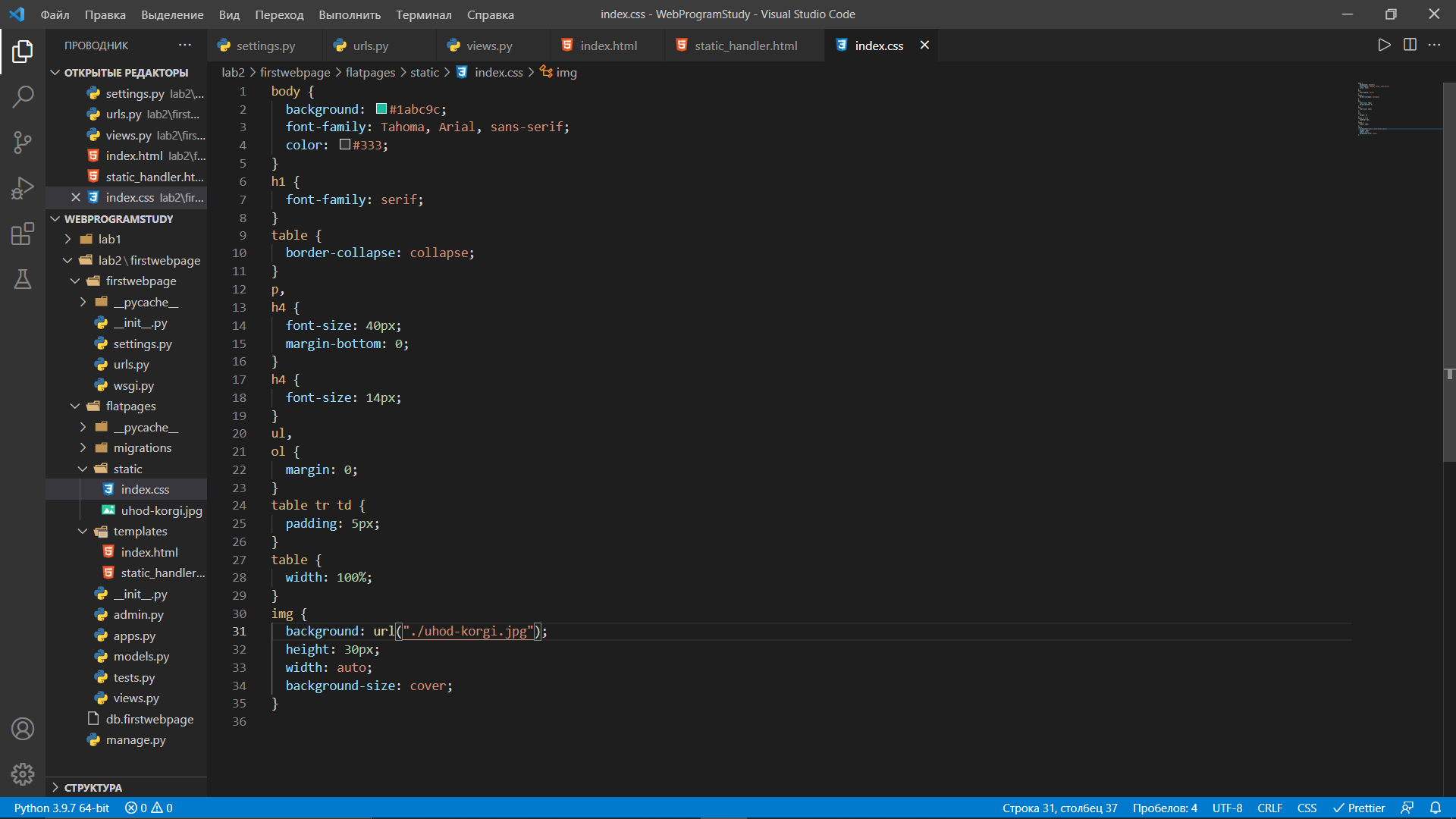


Рисунок 21 –css файл с выполненными заданиями

На рисунке 22 представлен результат.

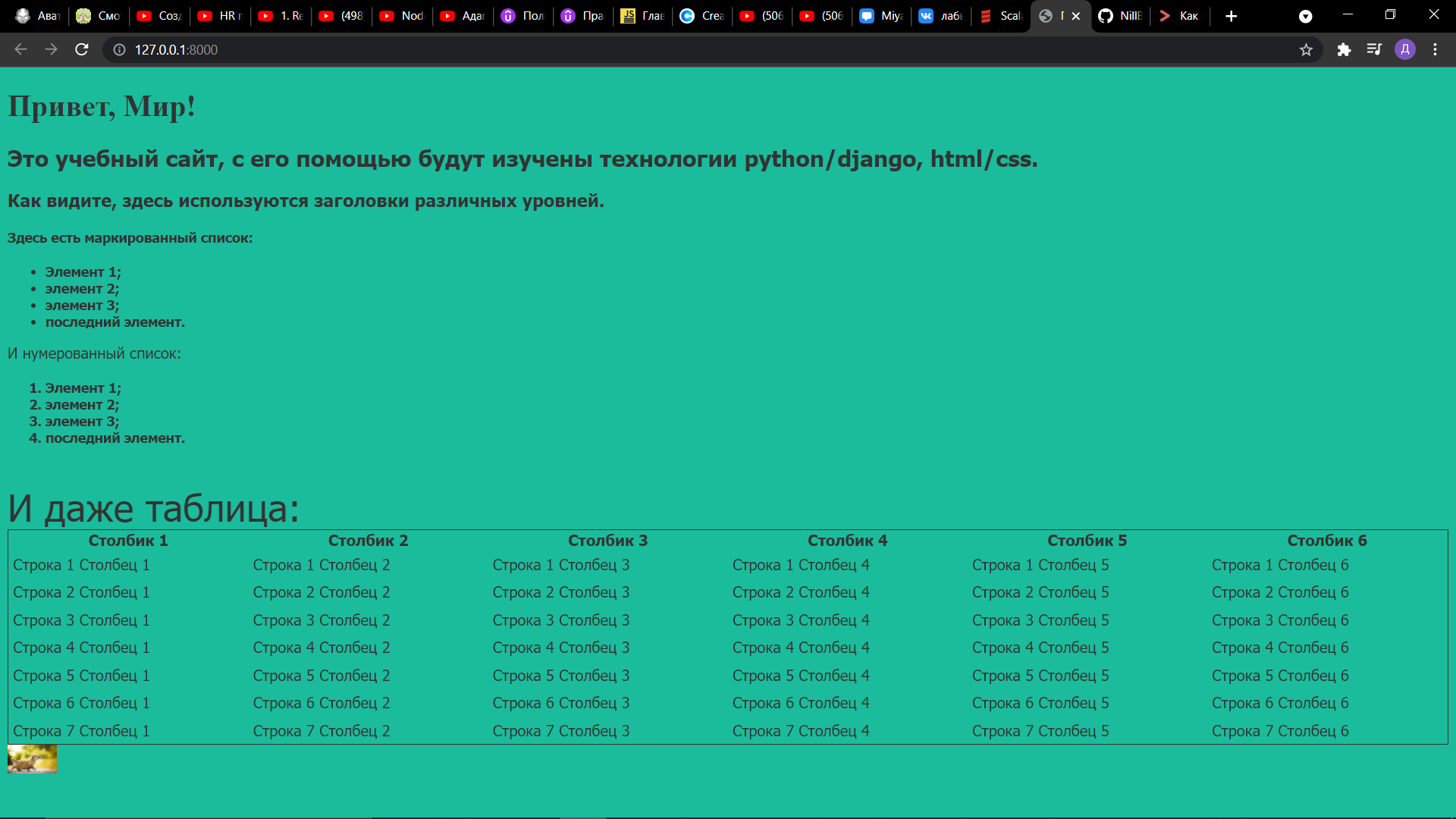


Рисунок 22 – результат работы

**Вывод:** я установил язык программирования «Python» и фреймворк Django, ознакомился с синтаксисом и выполнил задания.