**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 №5

«Создание формы и представления для нового поста»

по дисциплине:

«Web-программирование»

Выполнил:

Студент группы БФИ19

Бардюк Д.В.

Москва, 2021

**Задания:**

1. В директорию lab5 скопируйте ваш прошлый проект blog. В файле urls.py создайте ещё один адрес: article/new/, который будет обрабатываться функцией-представлением create\_post. Создайте в файле представлений наряду с функциями archive и get\_article новую функцию create\_post (на которую и ссылается только что написанный url). Эта функция должна будет при первом обращении к странице просто возвращать html-код с формой создания новой записи (html-шаблон для этого вы напишите чуть позже), а в случае, если к функции пользователь обратился во второй и следующий разы, при этом заполнив форму с названием и текстом нового поста, представление должно сохранять отправленные через форму данные в базу данных;
2. Создайте несколько статей через созданную форму;
3. Создайте стили, подключив CSS-файл к шаблону;
4. Добавьте проверку на то, что введенное название статьи уникально;
5. Загрузите ваш проект на любой гит-репозиторий (GitHub, GitLab, Google Code, Bitbucket и т.п.).

**2. Ход работы**

Создаём новую директорию для лабораторной работы №5 и копируем в неё проект blog из прошлой лабораторной работы. Добавим в файл urls.py новый путь, как показано на рисунке 1.

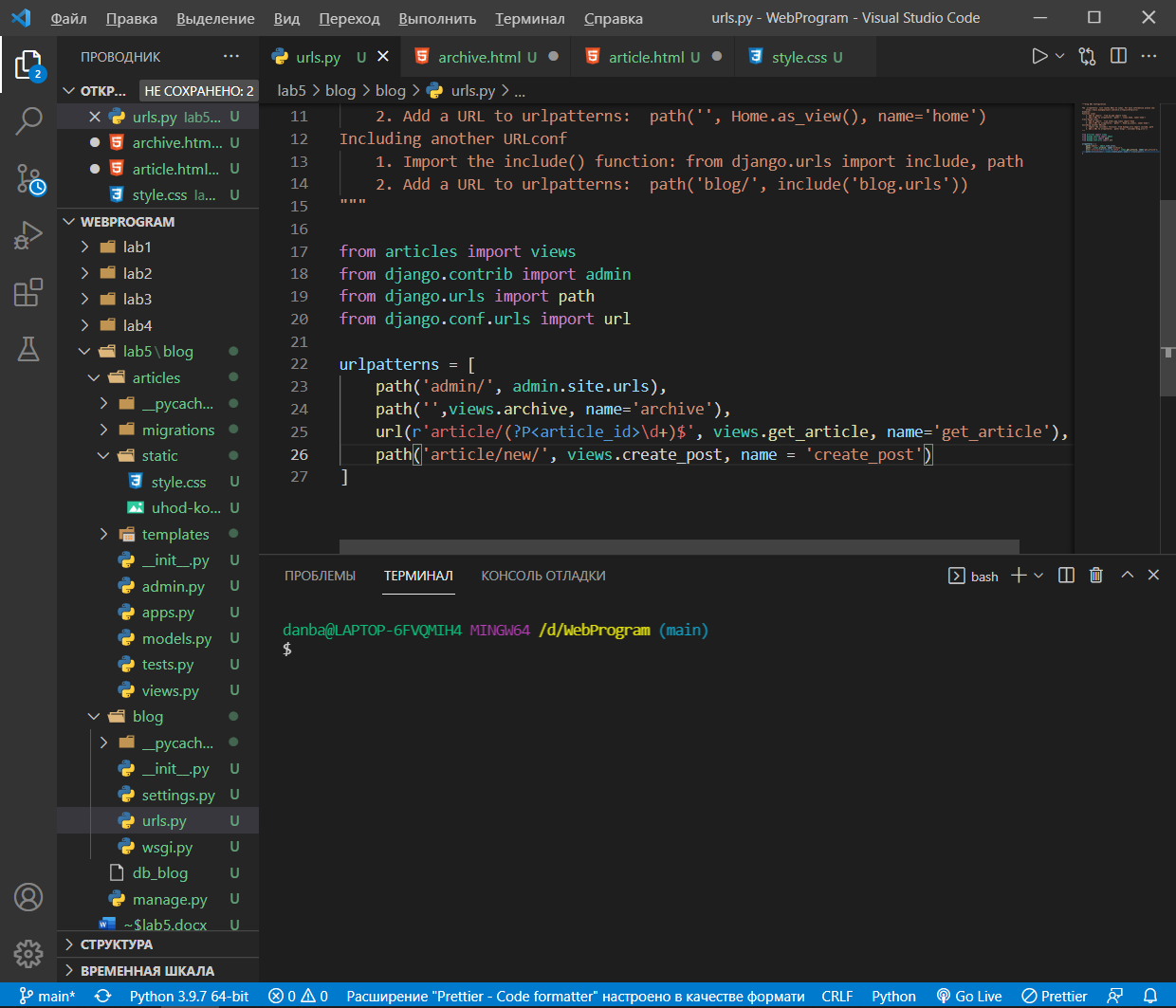


Рисунок 1 – Файл urls.py

Создадим в файле views.py новую функцию create\_post (рисунок 2) , которая должна будет при первом обращении к странице просто возвращать html-код с формой создания новой записи (html-шаблон для этого вы напишите чуть позже), а в случае, если к функции пользователь обратился во второй и следующий разы, при этом заполнив форму с названием и текстом нового поста, представление должно сохранять отправленные через форму данные в базу данных.

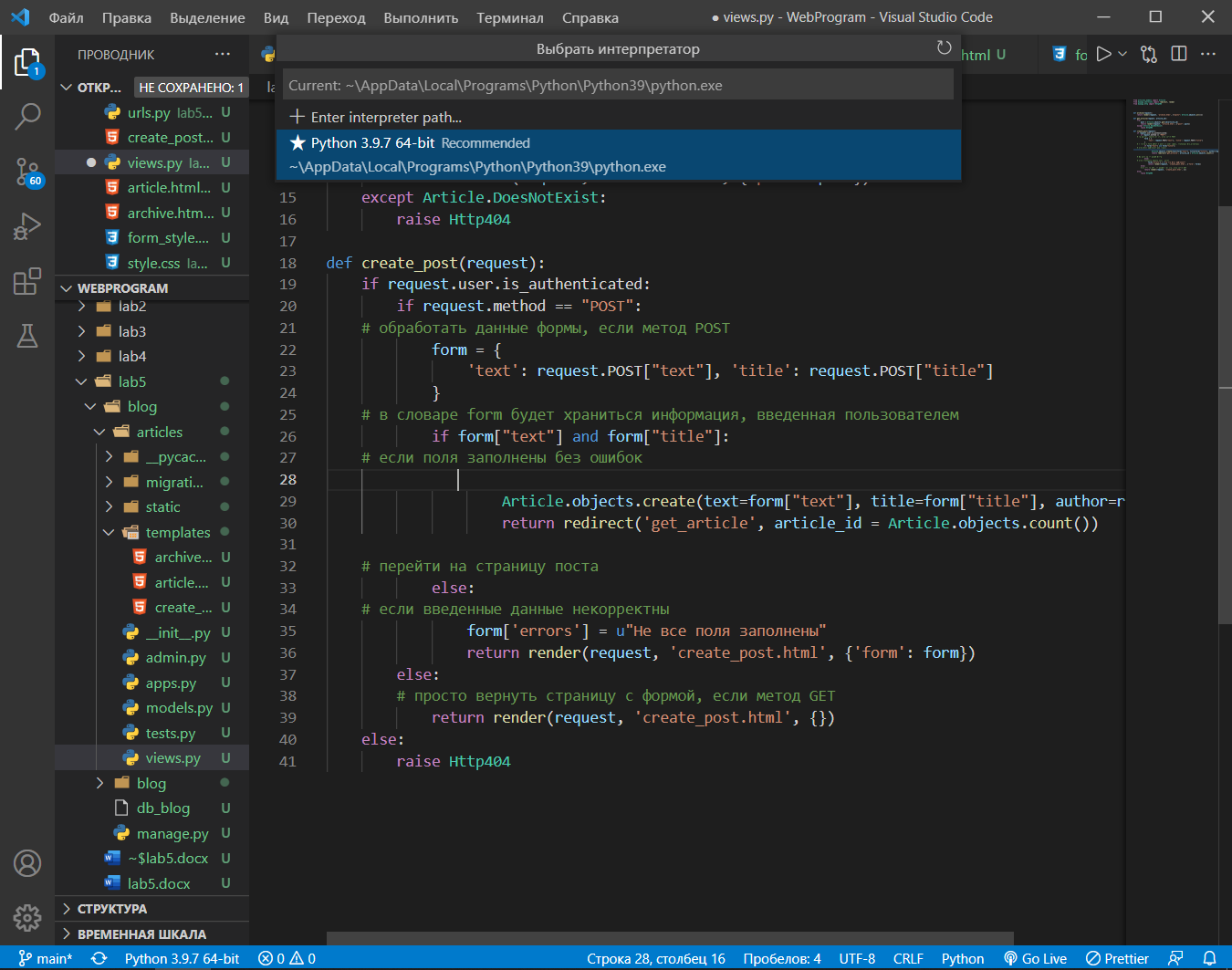


Рисунок 2 – Функция create\_post()

Создадим файл create\_post.html как на рисунке 3.

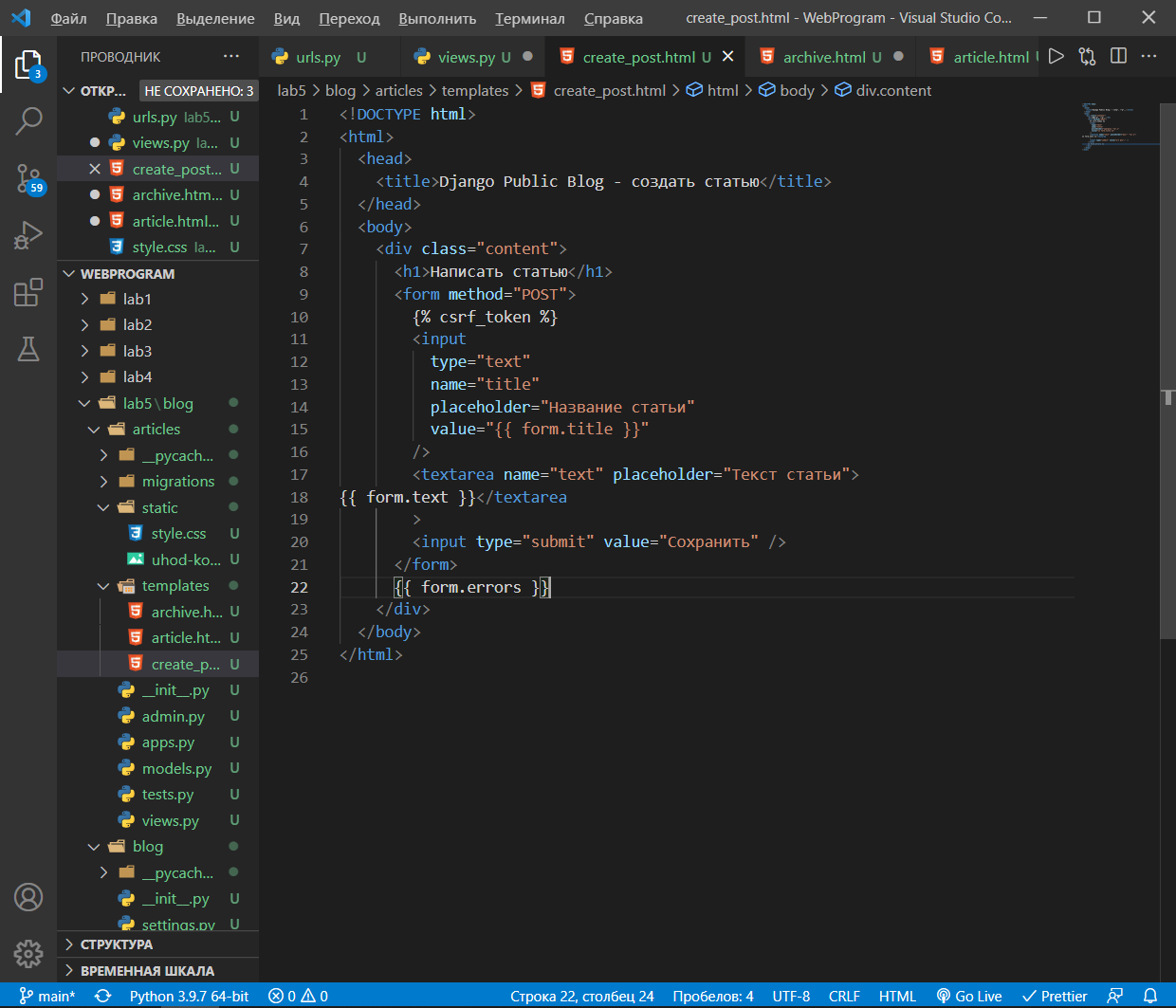


Рисунок 3 – Файл create\_post.html

Результат работы представлен на рисунке 4.

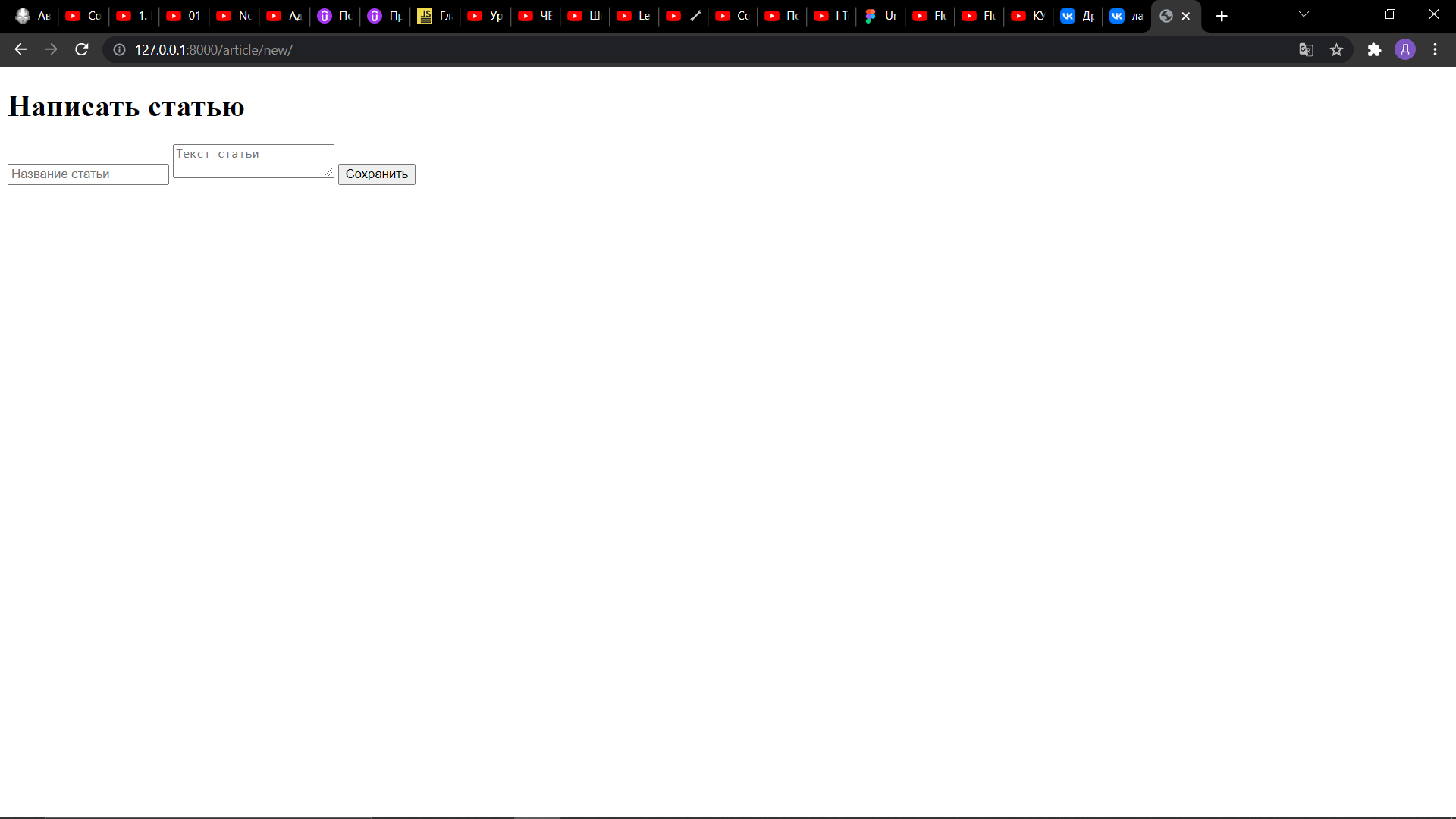


Рисунок 4 – Страница create\_post.html

Добавим статью через эту форму, результат будет представлен на рисунке 5.

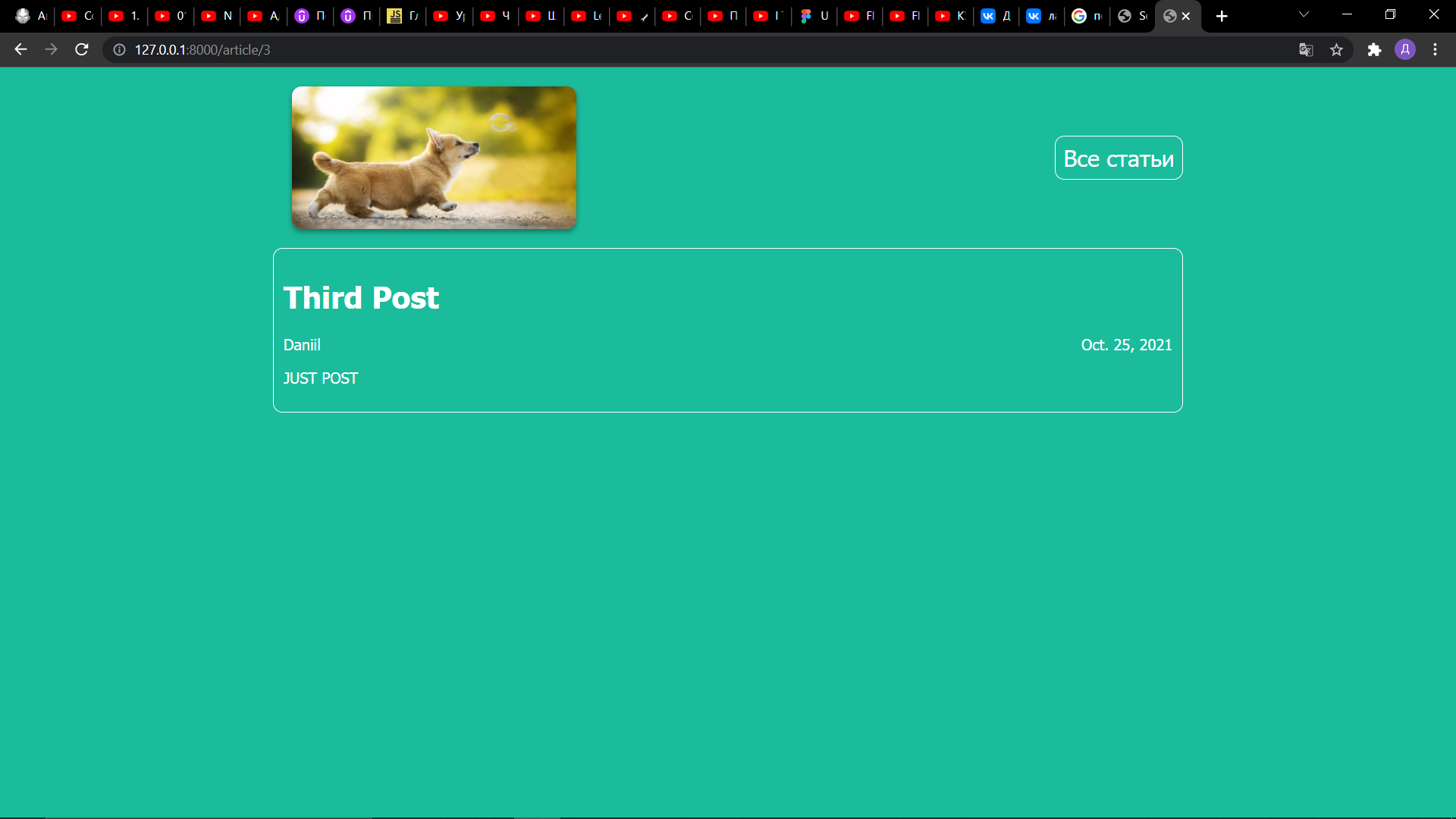


Рисунок 5 – Страница созданной статьи

Создадим файл со стилями для нашей страницы с формой создания статей как в листинге 1. Результат представлен на рисунке 6.

**Листинг 1**

body {

  background: #1abc9c;

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

  color: #ffffff;

  width: 960px;

  margin: 0 auto;

}

.content {

  background-color: rgba(133, 187, 173, 0.877);

  margin-top: 50px;

  padding: 20px;

  border-radius: 10px;

  border-color: white;

  border: 1px solid;

}

.content h1 {

  text-align: center;

}

form {

  height: 16rem;

  display: flex;

  flex-direction: column;

  justify-content: space-evenly;

  align-items: center;

}

input[type="text"] {

  height: 1, 2rem;

  width: 200px;

  text-align: center;

  border-radius: 10px;

  padding: 5px;

}

textarea {

  height: 100px;

  width: 300px;

  border-radius: 10px;

  padding: 5px;

}

input[type="submit"] {

  color: white;

  border-radius: 10px;

  height: 2rem;

  width: 8rem;

  padding: 5px;

  background-color: rgb(32, 182, 32);

}

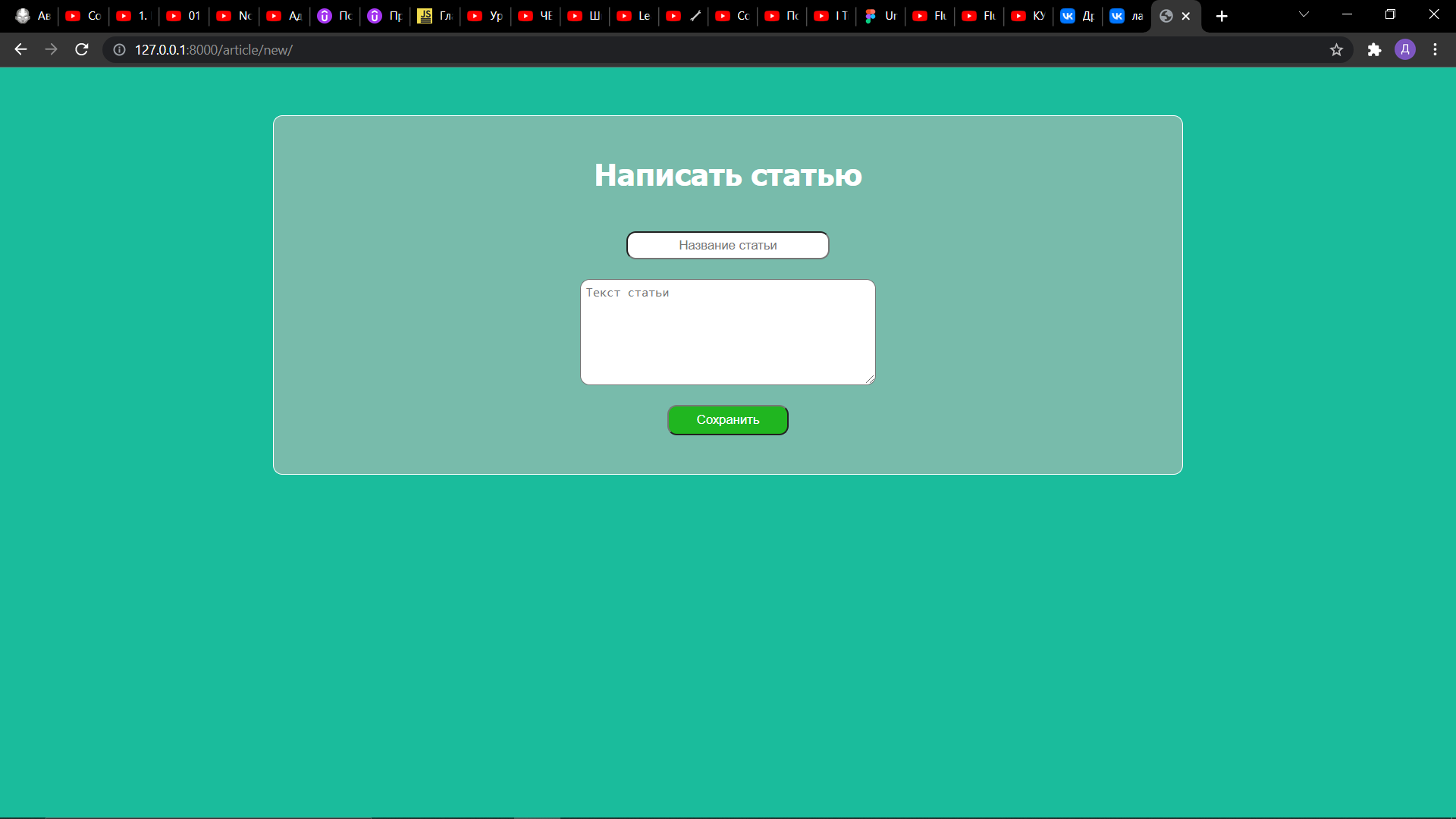


Рисунок 6 – Стилизованная страница с формой

Добавим в функцию create\_post() проверку на существование статьи с таким же заголовком как на рисунке 7. Результат работы будет показан на рисунке 8.

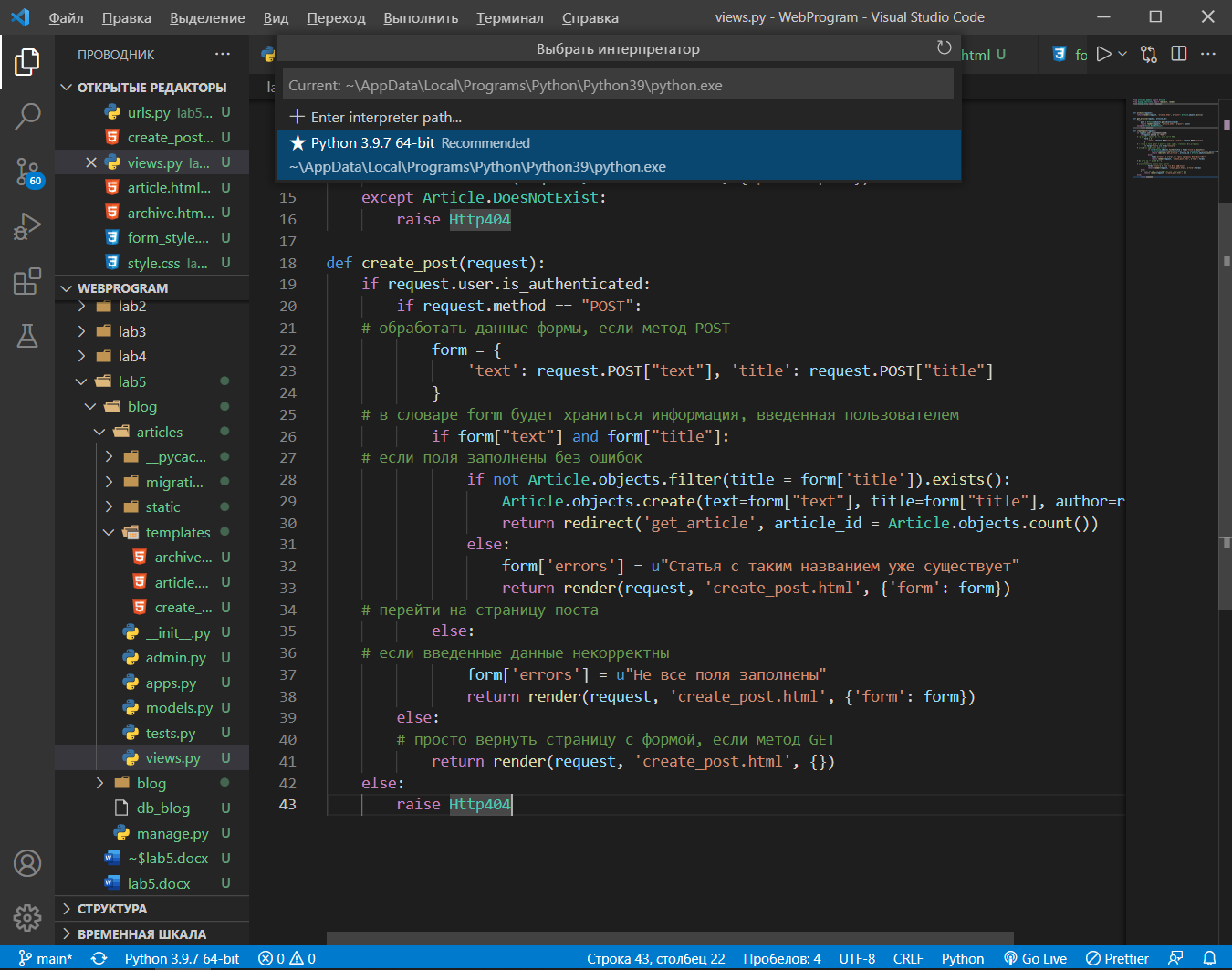


Рисунок 7 – Проверка на существование статей

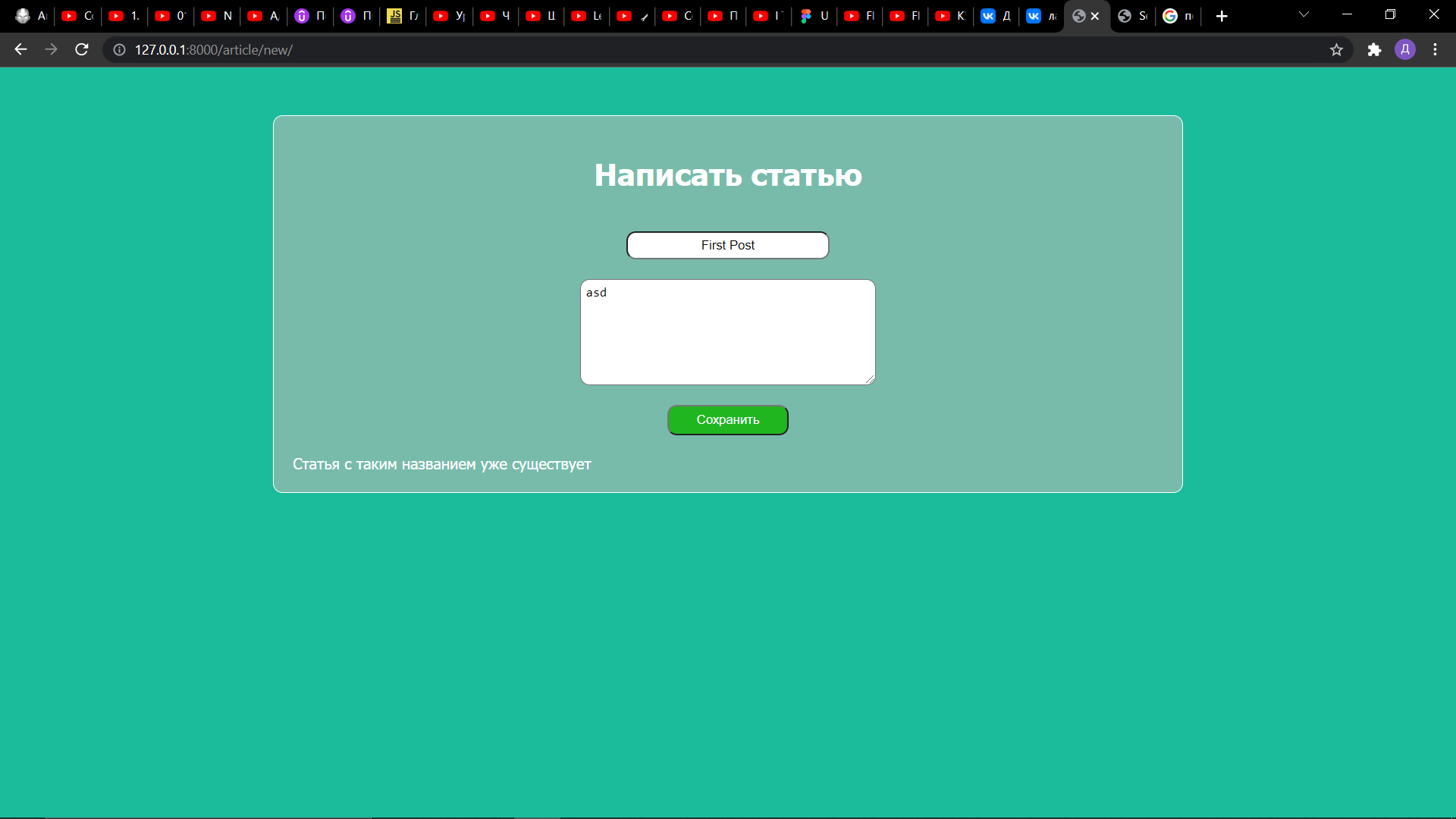


Рисунок 8 – Результат попытки добавление статьи с уже существующим названием

Для удобства использования, добавим на домашней странице ссылку на форму добавления статей как на рисунке 9.

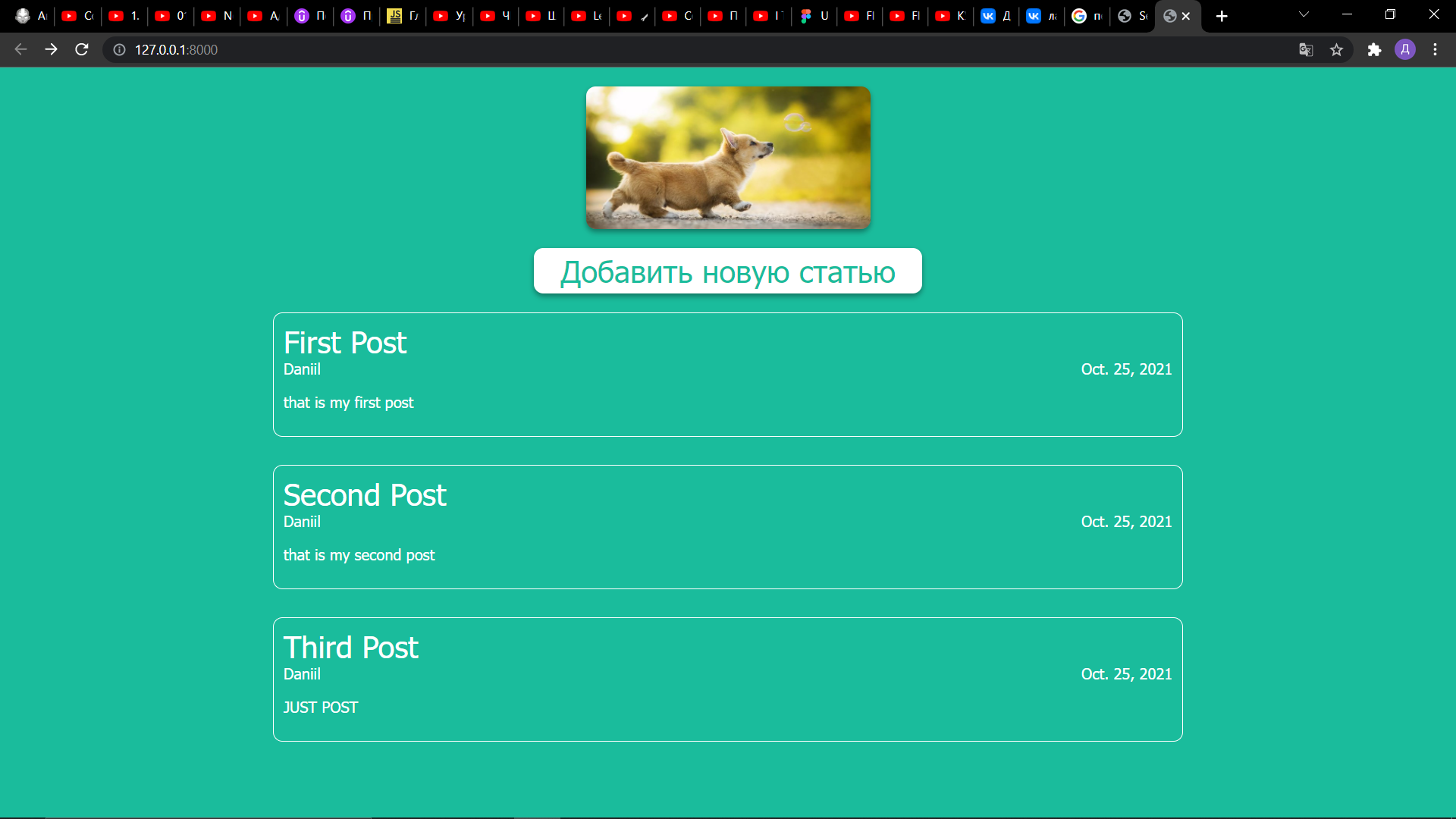


Рисунок 9 – Домашняя страница

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