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 №5

по дисциплине «Web-программирование»

Выполнил студент

группы БФИ1901

Соцков И.Н.

Проверил:

Москва, 2021

**Оглавление**

**[1 Задание на лабораторную работу](#_Toc28118)** [3](#_Toc28118)

**[2 Ход лабораторной работы](#_Toc28343)**[. 3](#_Toc28343)

[2.1 Создание формы и представления для нового поста 3](#_Toc32645)

**[Список используемых источников](#_Toc26553)** [10](#_Toc26553)

# **1 Задание на лабораторную работу**

1. Создайте несколько статей через созданную форму;

2. Создайте стили, подключив CSS-файл к шаблону;

3. Добавьте проверку на то, что введенное название статьи уникально;

# **2 Ход лабораторной работы.**

## 2.1 Создание формы и представления для нового поста

В файле urls.py создайте ещё один адрес: article/new/, который будет обрабатываться функцией-представлением create\_post. В представлении должна быть проверка на то, авторизован ли пользователь или нет. Если пользователь авторизован, то сначала определитесь, что программе необходимо сделать на данный момент: просто вернуть страницу с формой или пользователь в своё время уже получил форму, заполнил её и сейчас от программы требуется сохранить новую запись. Для этого нужно знать метод, с помощью которого отправлен запрос, и если это метод POST, то необходимо проверить полученные данные (если не заполнено одно из полей, вернуть обратно страницу с формой и уведомить об ошибке) и, если проверка пройдена, создать в базе данных новую запись. Также согласно заданию в конце методических указаний к лабораторной работе, необходимо создать проверку на то, является ли название статьи уникальным.

def create\_post(request):  
 if request.user.is\_authenticated:  
 if request.method == "POST":  
 form = {  
 'text': request.POST["text"], 'title': request.POST["title"]  
 }  
 if form["text"] and form["title"]:  
 if not Article.objects.filter(title=form['title']).exists():  
 Article.objects.create(text=form["text"], title=form["title"], author=request.user)  
 return redirect('get\_article', article\_id=Article.objects.count())  
 else:  
 form['errors'] = u"Статья с таким названием уже существует"  
 return render(request, 'create\_post.html', {'form': form})  
 else:  
  
 form['errors'] = u"Не все поля запонены"  
 return render(request, 'create\_post.html', {'form': form})  
 else:  
  
 return render(request, 'create\_post.html', {})  
 else:  
 raise Http404

Далее необходимо создать HTML-документ для отображение новой страницы на нашем сайте для создания новой статьи.

<!DOCTYPE html>  
<html>  
 <head>  
 <title>Django Public Blog - создать статью</title>  
 <link rel="stylesheet" href="{{ STATIC\_URL }}/static/article.css">  
 </head>  
 <body>  
 <div class="content">  
 <h1>Написать статью</h1>  
 <form class="create" method="POST">{% csrf\_token %}  
 <input type="text" name="title" placeholder="Название статьи" value="{{ form.title }}" maxlength="100">  
 <textarea name="text" placeholder="Текст статьи">{{ form.text }}</textarea>  
 <input type="submit" value="Сохранить" class="save\_button">  
 </form>  
 {{ form.errors }}  
 </div>  
 </body>  
</html>

Затем, создадим и подключим стили к новой странице на нашем сайте. Часть стилей возьмём из предыдущей лабораторной работы, а часть напишем сами для тегов «form», «input» и «textarea».

body{  
 background: #1abc9c;  
 font-family: Tahoma, Arial, sans-serif;  
 color: #ffffff  
}  
  
img {  
 display: block;  
 width: 318px;  
 margin-left: auto;  
 margin-right: auto;  
}  
  
.archive {  
 width: 960px;  
 margin-left: auto;  
 margin-right: auto;  
}  
  
.post-title a {  
 color: #ffffff;  
}  
  
.article-author {  
 width: 50%;  
 float: left;  
}  
  
.article-created-date {  
 text-align: right;  
}  
  
.article-image{  
 display: block;  
 width: 318px;  
 margin-left: 0;  
}  
  
.link {  
 color: white;  
 font-weight: bold;  
 position: absolute;  
 right: 470px;  
 top: 180px;  
}  
  
.article-border p{  
 text-align: right;  
}  
  
.article-text{  
 width: 960px;  
 text-align: justify;  
}  
.article-created-data{  
 text-align: right;  
}  
  
.content{  
 text-align: center;  
 padding-top: 70px;  
}  
  
input[name="title"]{  
 padding: 5px;  
 margin-bottom: 10px;  
 border: 1px solid #888;  
 outline: none;  
 -moz-appearance: none;  
 width: 200px;  
 text-align: center;  
 border-radius: 40px;  
}  
  
textarea[name="text"]{  
 padding: 25px;  
 margin-bottom: 10px;  
 border: 1px solid #888;  
 outline: none;  
 -moz-appearance: none;  
 width: 650px;  
 height: 350px;  
 resize: none;  
 border-radius: 40px;  
 scrollbar-width: thin;  
}  
  
.create{  
 display: flex;  
 flex-direction: column;  
 align-items: center;  
}  
  
.save\_button{  
 padding: 10px;  
 width: 150px;  
 background-color: white;  
 border: none;  
 border-radius: 40px;  
 color: #1abc9c;  
 font-weight: bold;  
 letter-spacing: 0.06em;  
 margin-top: 10px;  
}  
  
.save\_button:hover{  
 color: white;  
 background-color: #1abc9c;  
 box-shadow: 1px 1px 10px 10px;  
 transition-duration: 0.3s;  
}

Запустим сервер и перейдём по адресу <http://127.0.0.1:8000/article/new/>. После перехода , должно появиться новая страница, которую мы создали. Внешний вид новой страницы показан на рисунке 1.

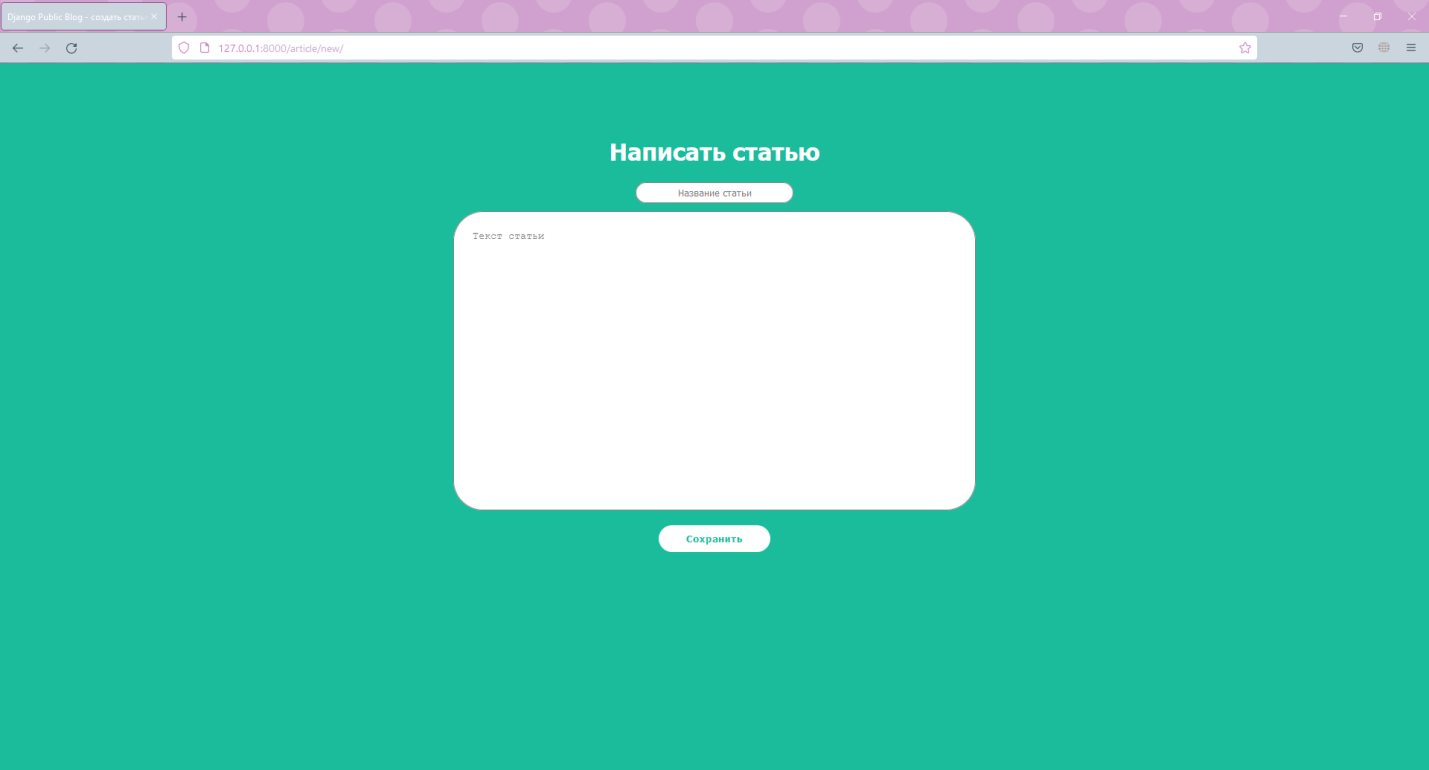
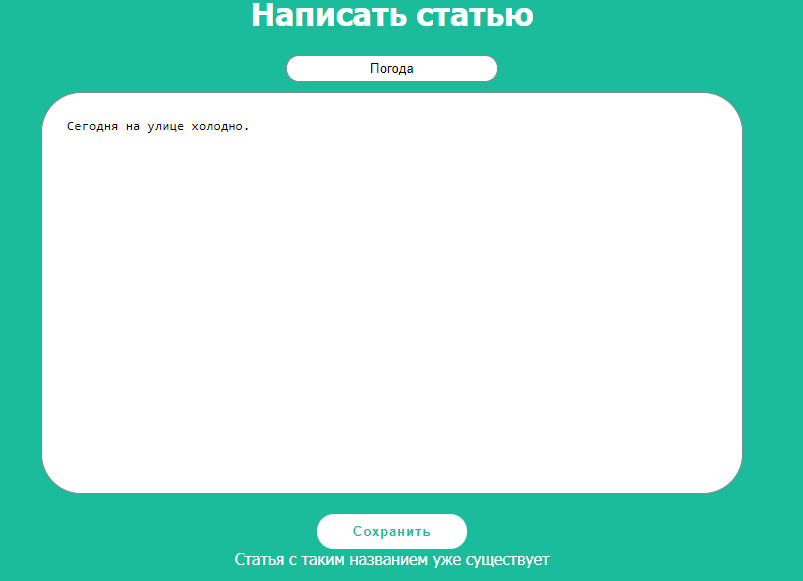


Рисунок 1 – Страницы создания статьи

Далее, попробуем ввести название статьи, которая уже была когда-то создана. Как мы видим на рисунке 2, на странице появляется сообщение о том что статься с таким названием уже существует, значит все работает верно.

Рисунок 2 – Ошибка в случае того, что статья с таким названием уже существует

А теперь, попробуем создать новую статью название и текст которой не похожи на название и тест предыдущих статей. Назовём статью «МТУСИ» и в тексте напишем «Московский Технический Университет Связи и Информатики». На рисунке 3 можно увидеть создание данной статьи.

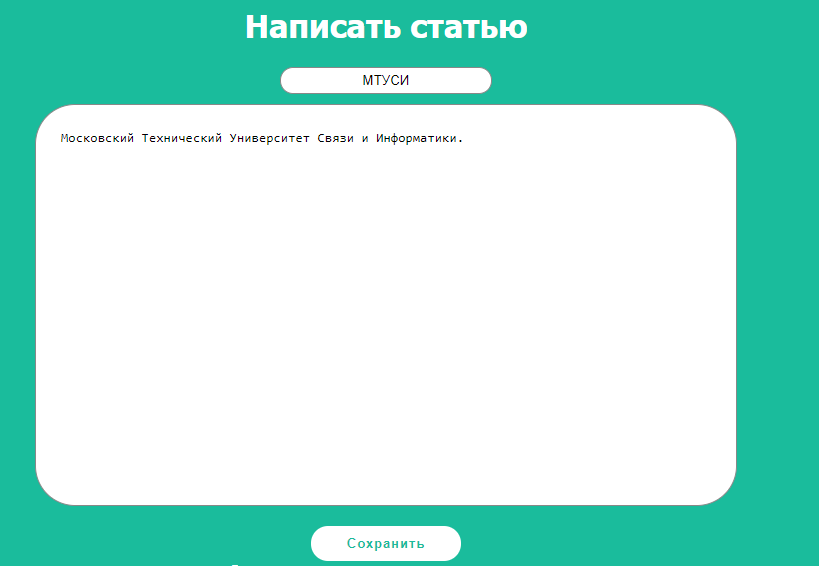


Рисунок 3 - Создание новой статьи

После того как мы нажмём на кнопку «Сохранить» перед нами откроется страница статьи, что подтверждает её успешное создание.



Рисунок 4 - Успешно созданная статья

В завершении лабораторной работы, загрузим файлы на GitHub. На рисунке 5 показана успешная загрузка файлов.

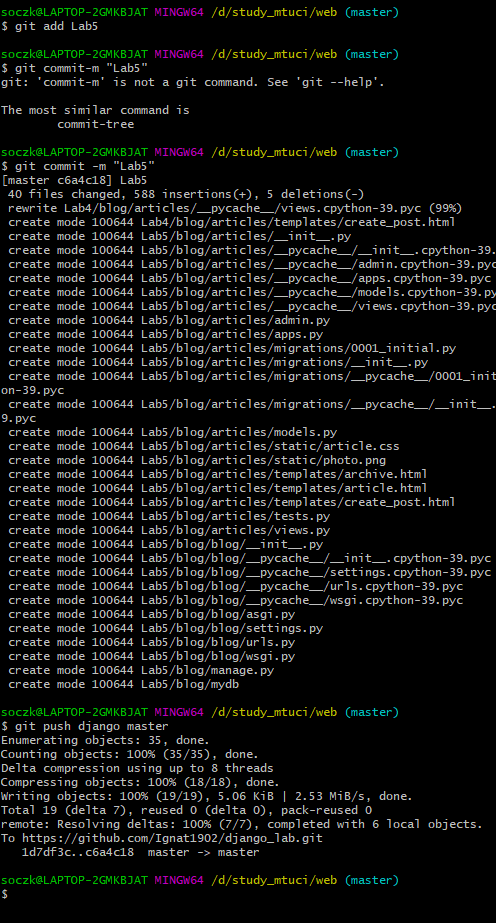


Рисунок 5 – Загрузка проекта на GitHub

**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.