Федеральное агентство связи

Орден Трудового Красного Знамени федерального государственного бюджетного образовательного учреждения высшего образования «Московский технический университет связи и информатики»

Кафедра «Математической кибернетики и информационных технологий»

Лабораторные работы по дисциплине «Кроссплатформенные технологии программирования»

Лабораторная работа №7

«Создание формы и представления для нового поста»

Выполнил студент:

Группа БСТ-1703

Тешаев Парвиз

Москва 2019

**Среда разработки: Visual Studio Code**

**Задание:**

* Создайте стили, подключив css-файл к шаблону, которые соответствуют макету lab5\_creation\_form.
* Добавьте проверку на то, что введенное для нового поста название уникально.

Вид всех статей (Рисунок 1):

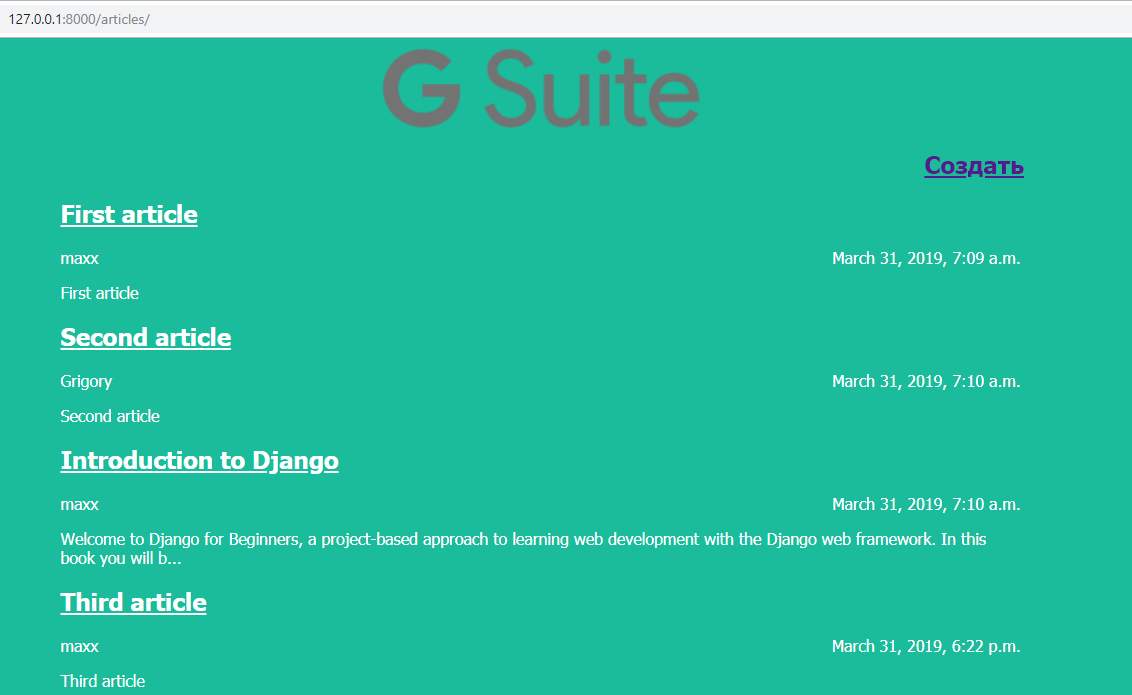


Рисунок 1 – Все статьи на сайте

Вид каждой статьи (Рисунок 2):

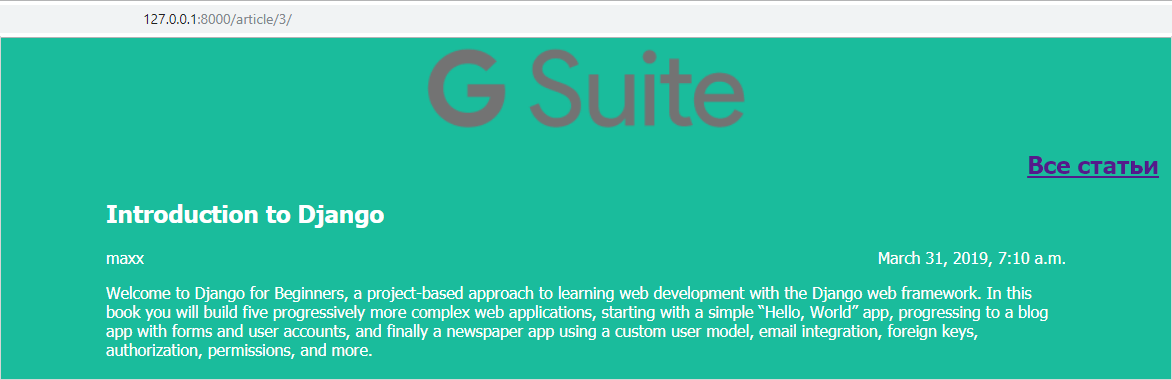


Рисунок – Отображение конкретной статьи на сайте

Вид создания новой статьи (Рисунок 3):

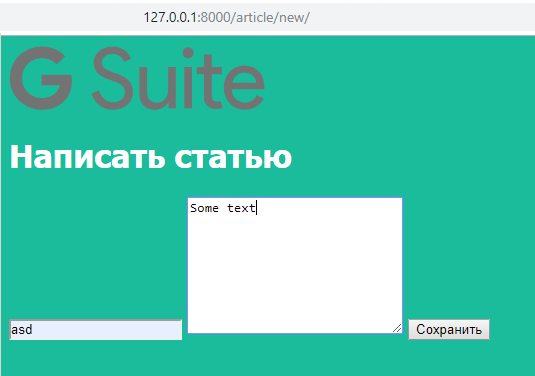


Рисунок – Страница создания новой статьи

Код проекта:

**#blog/urls.py**

from django.contrib import admin

from django.urls import path, include

urlpatterns = [

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

]

urlpatterns += [

path('articles/', include('articles.urls')),

]

**#articles/urls.py**

from django.urls import path

from articles import views

urlpatterns = [

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

path('article/new/', views.create\_post, name='create\_post'),

path('article/<article\_id>/', views.get\_article, name='get\_article'),

]

**#articles/views.py**

from .models import Article

from django.shortcuts import render

def archive(request):

return render(request, 'archive.html', {"posts":Article.objects.all()})

def get\_article(request, article\_id):

try:

post = Article.objects.get(id=article\_id)

return render(request, 'article.html', {"post": post})

except Article.DoesNotExist:

raise Http404

def create\_post(request):

if request.user.is\_authenticated:

if request.method == "POST":

# обработать данные формы, если метод POST

form = {

'text': request.POST["text"],

'title': request.POST["title"]

}

# в словаре form будет храниться информация, введенная пользователем

if form["text"] and form["title"]:

# если поля заполнены без ошибок

if not Article.objects.filter(title=form['title']):

# если заголовок уникален

Article.objects.create(text=form["text"], title=form["title"], author=request.user)

article = Article.objects.get(title=form['title'])

return redirect('get\_article', article\_id=article.id)

# перейти на страницу поста

else:

# если заголовок не уникален

form['errors'] = u"Заголовок не уникален"

return render(request, 'create\_post.html', {'form': form})

else:

# если введенные данные некорректны

form['errors'] = u"Не все поля заполнены"

return render(request, 'create\_post.html', {'form': form})

else:

# просто вернуть страницу с формой, если метод GET

return render(request, 'create\_post.html', {})

else:

raise Http404

**#articles/models.py**

from django.conf import settings

from django.db import models

class Article(models.Model):

objects = models.Manager()

author = models.ForeignKey(settings.AUTH\_USER\_MODEL, on\_delete=models.CASCADE)

title = models.CharField(max\_length=200)

text = models.TextField()

created\_date = models.DateTimeField(default=timezone.now)

def \_\_str\_\_(self):

return self.title

def get\_excerpt(self):

return self.text[:140] + "..." if len(self.text) > 140 else self.text

**articles/templates/archive.html**

<title> Архив всех статей </title>

<body>

<div class="header">

{% load static %}

<img src="{% static 'img/dpb-logo.png' %}">

<h2 class="exit"><a href="/article/new">Создать</a></h2>

<link rel="stylesheet" href="{% static 'css/article.css' %}">

</div>

<div class="archive">

{% for post in posts %}

<div class="one-post">

<h2 class="post-title">

<a href="/article/{{ post.id }}">{{ post.title }}</a>

</h2>

<div class="article-info">

<div class="article-author">{{ post.author.username }}</div>

<div class="article-created-date">{{ post.created\_date }}</div>

</div>

<p class="article-text">{{ post.get\_excerpt }}</p>

</div>

{% endfor %}

</div>

</body>

**articles/templates/article.html**

<title> {{post.title}} </title>

<body>

<div class="header">

{% load static %}

<img src="{% static 'img/dpb-logo.png' %}">

<h2 class="exit"><a href="/articles/">Все статьи</a></h2>

<link rel="stylesheet" href="{% static 'css/article.css' %}">

</div>

<div class="article">

<div class="one-post">

<h2 class="post-title">{{ post.title }}</h2>

<div class="article-info">

<div class="article-author">{{ post.author.username }}</div>

<div class="article-created-date">{{ post.created\_date }}</div>

</div>

</div>

<p class="article-text">{{ post.text }}</p>

</div>

</body>

**articles/templates/create\_post.html**

<html>

<head>

<title>Django Public Blog - создать статью</title>

{% load static %}

<img src="{% static 'img/dpb-logo.png' %}">

<link rel="stylesheet" href="{% static 'css/article.css' %}">

</head>

<body>

<div class="content">

<h1>Написать статью</h1>

<form method="POST">{% csrf\_token %}

<input type="text" name="title" placeholder="Название статьи" value="{{ form.title }}">

<textarea name="text" placeholder="Текст статьи">{{ form.text }}</textarea>

<input type="submit" value="Сохранить">

</form>

{{ form.errors }}

</div>

</body>

</html>

**articles/static/css/article.css**

body {

background: #1abc9c;

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

color: #ffffff;

}

.header 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 {

width: 960px;

margin-left: auto;

margin-right: auto;

}

.exit {

width: 1150px;

text-align: right;

}