

**МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ СВЯЗИ И МАССОВЫХ  
КОММУНИКАЦИЙ**

**Ордена Трудового Красного Знамени**

**Федеральное государственное бюджетное образовательное учреждение  
высшего образования**

**«Московский технический университет связи и информатики»**

Кафедра «Математической кибернетики и информационных технологий»

**Отчет по лабораторной работе №5**

по дисциплине «Введение в информационные технологии» на тему:

Создание формы и представления для нового поста

Выполнил:

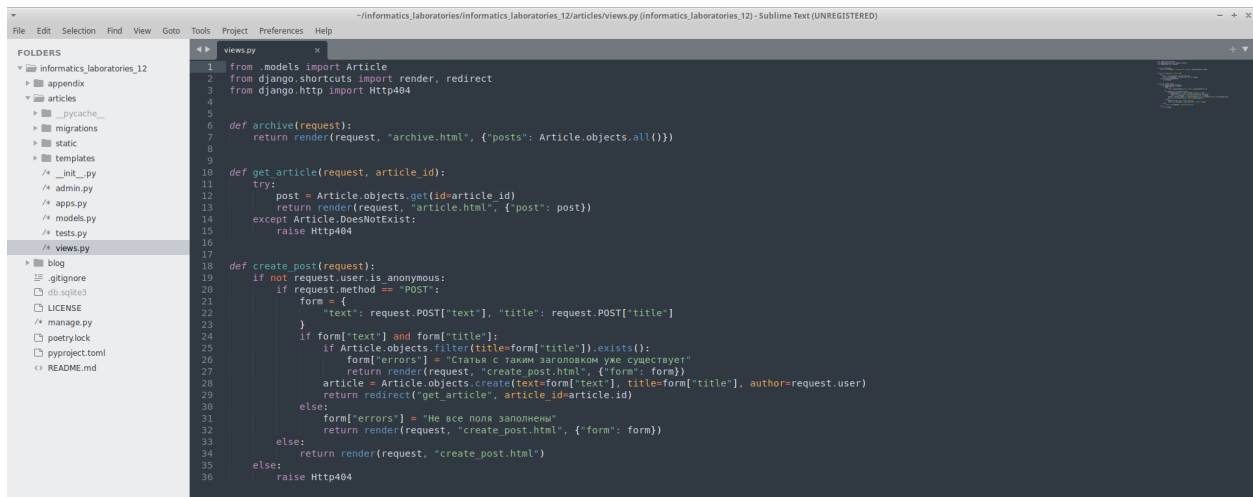
Сидорук Данил Вадимович, БФИ-2202

Проверил: Фурлетов Ю. М.

Москва

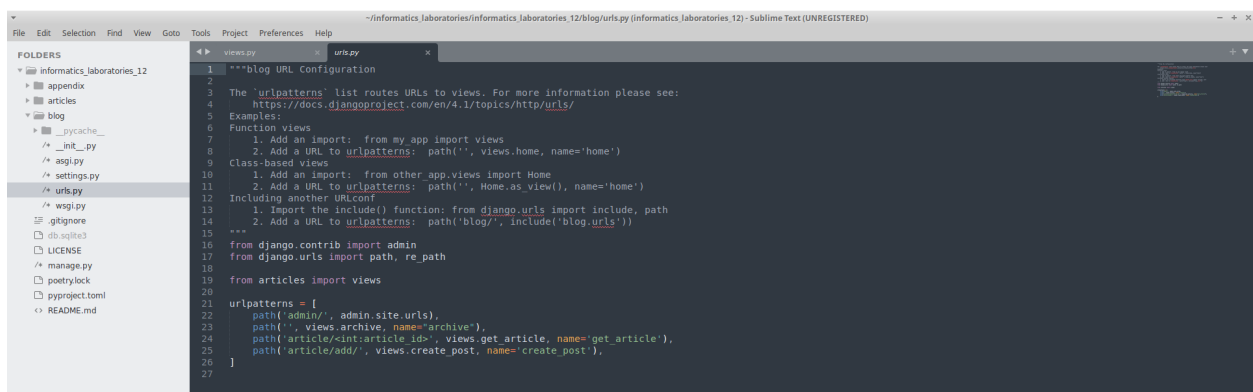
2023

На рисунке ниже приведен обработчик создания нового поста



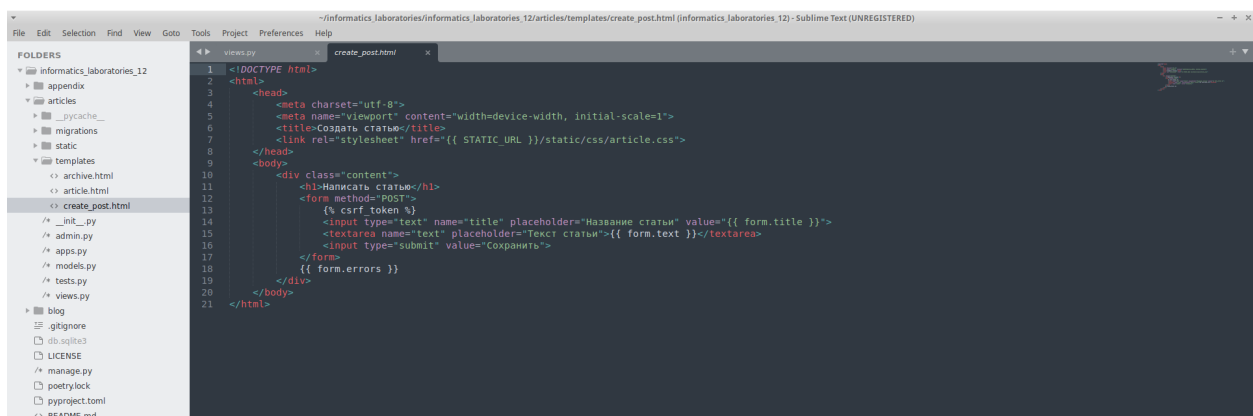
```
1 from .models import Article
2 from django.shortcuts import render, redirect
3 from django.http import Http404
4
5
6 def archive(request):
7     return render(request, "archive.html", {"posts": Article.objects.all()})
8
9
10 def get_article(request, article_id):
11     try:
12         post = Article.objects.get(id=article_id)
13         return render(request, "article.html", {"post": post})
14     except Article.DoesNotExist:
15         raise Http404
16
17
18 def create_post(request):
19     if not request.user.is_anonymous:
20         if request.method == "POST":
21             form = {
22                 "text": request.POST["text"], "title": request.POST["title"]
23             }
24             if form["text"] and form["title"]:
25                 if Article.objects.filter(title=form["title"]).exists():
26                     form["errors"] = "Сравна с таким заголовком уже существует"
27                     return render(request, "create_post.html", {"form": form})
28                 article = Article.objects.create(text=form["text"], title=form["title"], author=request.user)
29                 return redirect("get_article", article_id=article.id)
30             else:
31                 form["errors"] = "Не все поля заполнены"
32                 return render(request, "create_post.html", {"form": form})
33         else:
34             return render(request, "create_post.html")
35     else:
36         raise Http404
```

На рисунке ниже приведен urlpattern для данного обработчика



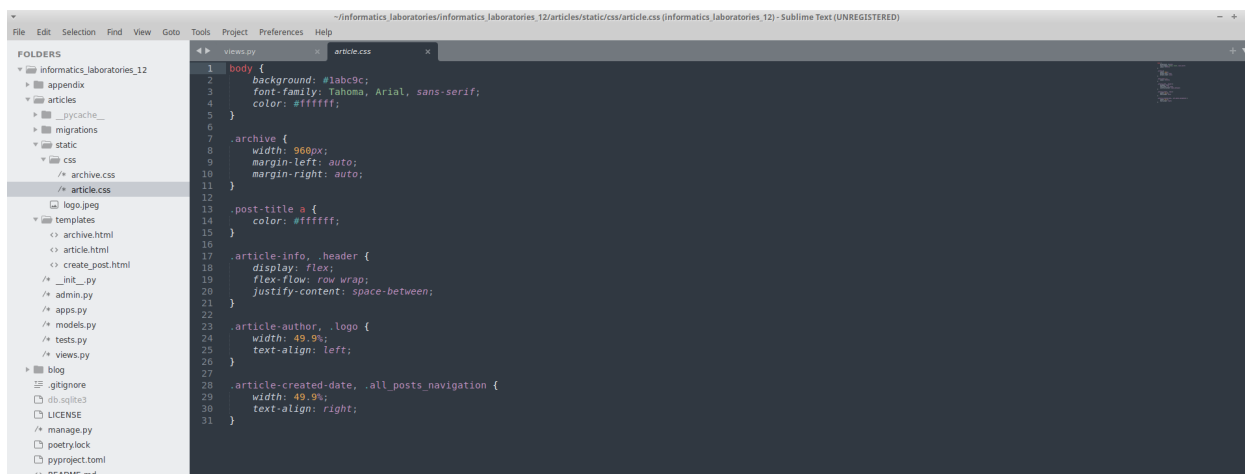
```
1 """blog URL Configuration
2
3 The 'urlpatterns' list routes URLs to views. For more information please see:
4 https://docs.djangoproject.com/en/4.1/topics/http/urls/
5 Examples:
6 Function views
7 1. Add an import: from my_app import views
8 2. Add a URL to urlpatterns: path('', views.home, name='home')
9 Class-based views
10 1. Add an import: from other_app.views import Home
11 2. Add a URL to urlpatterns: path('', Home.as_view(), name='home')
12 Including another URLconf
13 1. Import the include() function: from django.urls import include, path
14 2. Add a URL to urlpatterns: path('blog/', include('blog.urls'))
15 """
16 from django.contrib import admin
17 from django.urls import path, re_path
18 from articles import views
19
20 urlpatterns = [
21     path('admin/', admin.site.urls),
22     path('', views.archive, name='archive'),
23     path('article/<int:article_id>', views.get_article, name='get_article'),
24     path('article/add/', views.create_post, name='create_post'),
25 ]
```

На рисунке ниже приведен шаблон страницы создания статьи

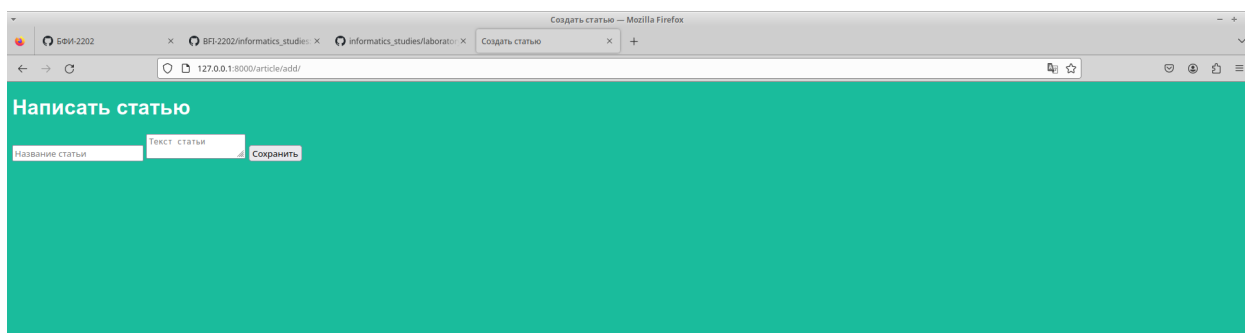


```
1 <!DOCTYPE html>
2 <html>
3 <head>
4     <meta charset="utf-8">
5     <meta name="viewport" content="width=device-width, initial-scale=1">
6     <title>Создать статью</title>
7     <link rel="stylesheet" href="{% STATIC_URL %}/static/css/article.css">
8 </head>
9 <body>
10     <div class="content">
11         <h1>Написать статью</h1>
12         <form method="POST">
13             {% csrf_token %}
14             <input type="text" name="title" placeholder="Название статьи" value="{% form.title %}">
15             <textarea name="text" placeholder="Текст статьи">{% form.text %}</textarea>
16             <input type="submit" value="Сохранить">
17         </form>
18         {% form.errors %}
19     </div>
20 </body>
21 </html>
```

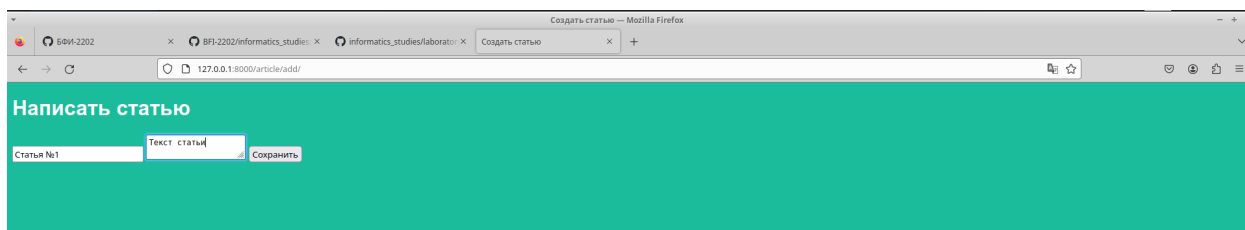
На рисунке ниже приведен файл стилей, используемый в шаблоне, представленном выше



На рисунке приведен вид страницы создания статьи



На рисунке ниже приведен пример заполнения формы создания статьи



На рисунке ниже приведен результат нажатия на кнопку «отправить» при заполненной форме создания статьи

