

**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 №6**

по дисциплине «Введение в информационные технологии» на тему:

Создание формы, шаблона и представления для регистрации и авторизации

Выполнил:

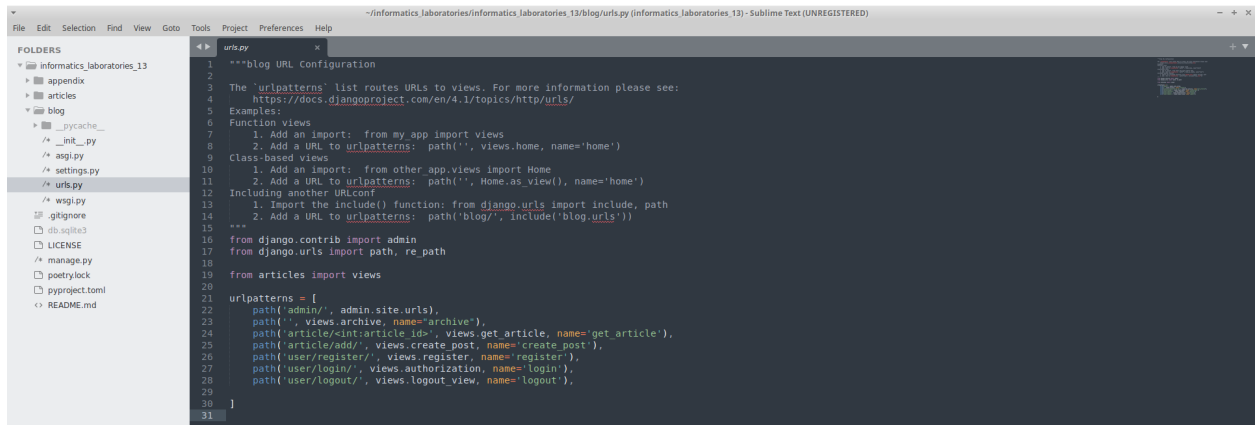
Сидорук Данил Вадимович, БФИ-2202

Проверил: Фурлетов Ю. М.

Москва

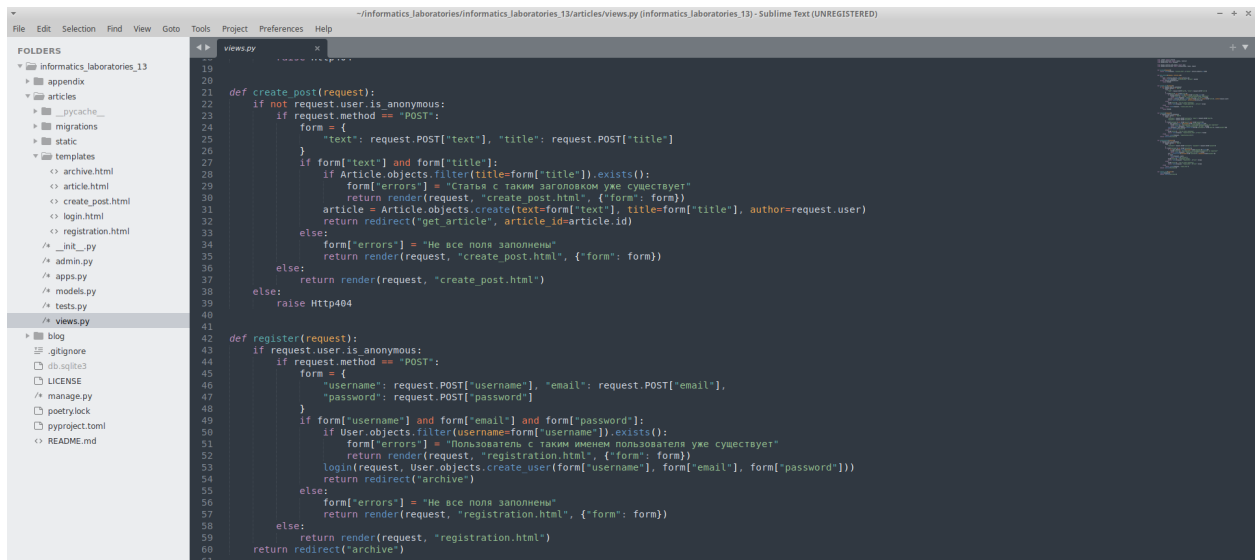
2023

На рисунке ниже приведены urlpatterns для всех обработчиков



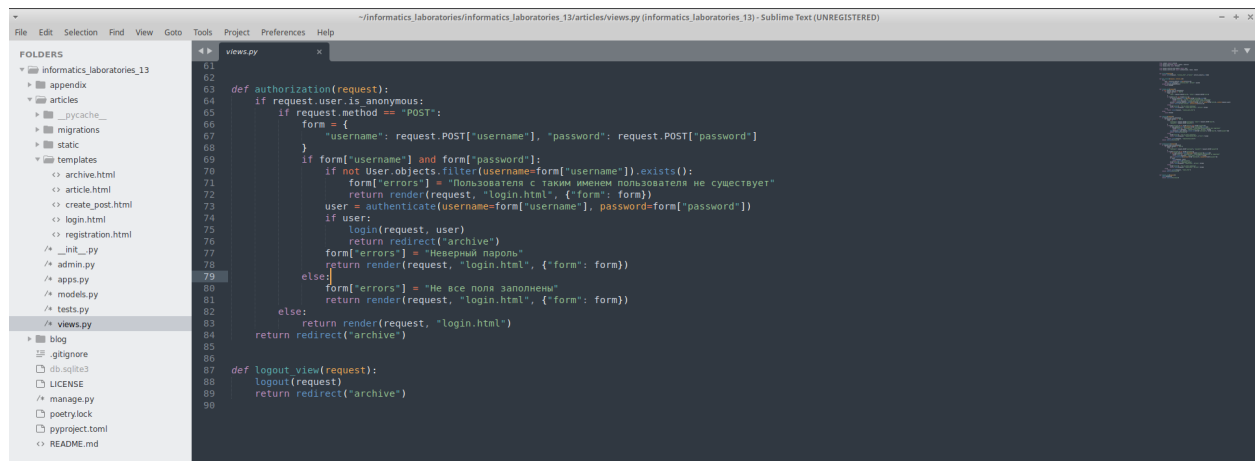
```
1  """blog URL Configuration
2
3  The 'urlpatterns' list routes URLs to views. For more information please see:
4      https://docs.djangoproject.com/en/4.1/topics/http/urls/
5  Examples:
6  Function views
7      1. Add an import: from my_app import views
8      2. Add a URL to urlpatterns: path('', 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     path('user/register/', views.register, name='register'),
26     path('user/login/', views.authorization, name='login'),
27     path('user/logout/', views.logout_view, name='logout'),
28 ]
29
30
31
```

На рисунке ниже приведены обработчики create\_post и register



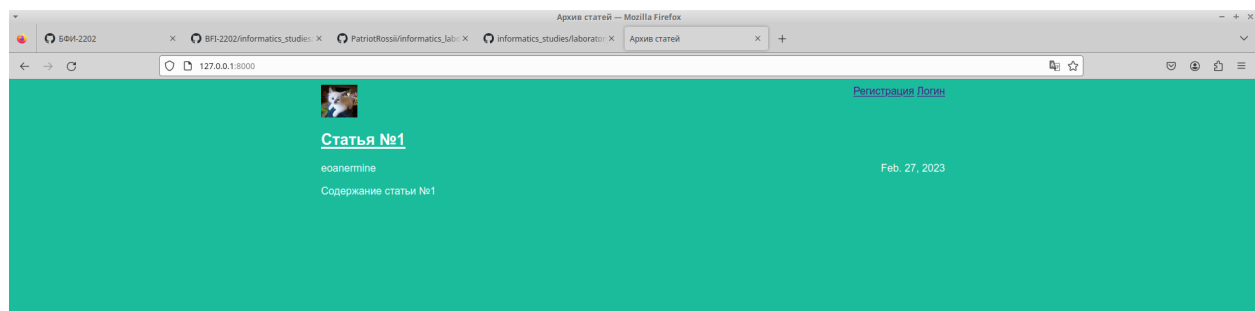
```
19
20
21 def create_post(request):
22     if not request.user.is_anonymous:
23         if request.method == "POST":
24             form = {
25                 "text": request.POST["text"], "title": request.POST["title"]
26             }
27             if form["text"] and form["title"]:
28                 if Article.objects.filter(title=form["title"]).exists():
29                     form["errors"] = "Статья с таким заголовком уже существует"
30                     return render(request, "create_post.html", {"form": form})
31                 article = Article.objects.create(text=form["text"], title=form["title"], author=request.user)
32                 return redirect('get_article', article_id=article.id)
33             else:
34                 form["errors"] = "He ace none заголовка"
35                 return render(request, "create_post.html", {"form": form})
36         else:
37             return render(request, "create_post.html")
38     else:
39         raise Http404
40
41
42 def register(request):
43     if request.user.is_anonymous:
44         if request.method == "POST":
45             form = {
46                 "username": request.POST["username"], "email": request.POST["email"],
47                 "password": request.POST["password"]
48             }
49             if form["username"] and form["email"] and form["password"]:
50                 if User.objects.filter(username=form["username"]).exists():
51                     form["errors"] = "Пользователь с таким именем пользователя уже существует"
52                     return render(request, "registration.html", {"form": form})
53                 login(request, User.objects.create_user(form["username"], form["email"], form["password"]))
54                 return redirect('archive')
55             else:
56                 form["errors"] = "He ace none заголовка"
57                 return render(request, "registration.html", {"form": form})
58         else:
59             return render(request, "registration.html")
60     return redirect('archive')
61
```

На рисунке ниже приведены обработчики authorization и logout\_view

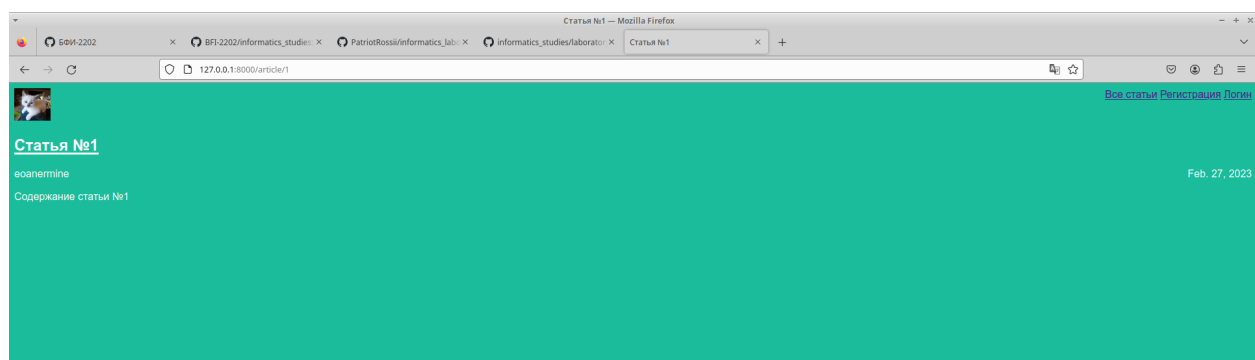


```
61
62
63 def authorization(request):
64     if request.user.is_anonymous:
65         if request.method == 'POST':
66             form = {
67                 "username": request.POST["username"], "password": request.POST["password"]
68             }
69             if form["username"] and form["password"]:
70                 if not User.objects.filter(username=form["username"]).exists():
71                     form["errors"] = "Пользователя с таким именем пользователя не существует"
72                     return render(request, "login.html", {"form": form})
73                 user = authenticate(username=form["username"], password=form["password"])
74                 if user:
75                     login(request, user)
76                     return redirect("archive")
77                 form["errors"] = "Неверный пароль"
78                 return render(request, "login.html", {"form": form})
79             else:
80                 form["errors"] = "Не все поля заполнены"
81                 return render(request, "login.html", {"form": form})
82             else:
83                 return render(request, "login.html")
84             return redirect("archive")
85
86
87
88 def logout_view(request):
89     logout(request)
90     return redirect("archive")
```

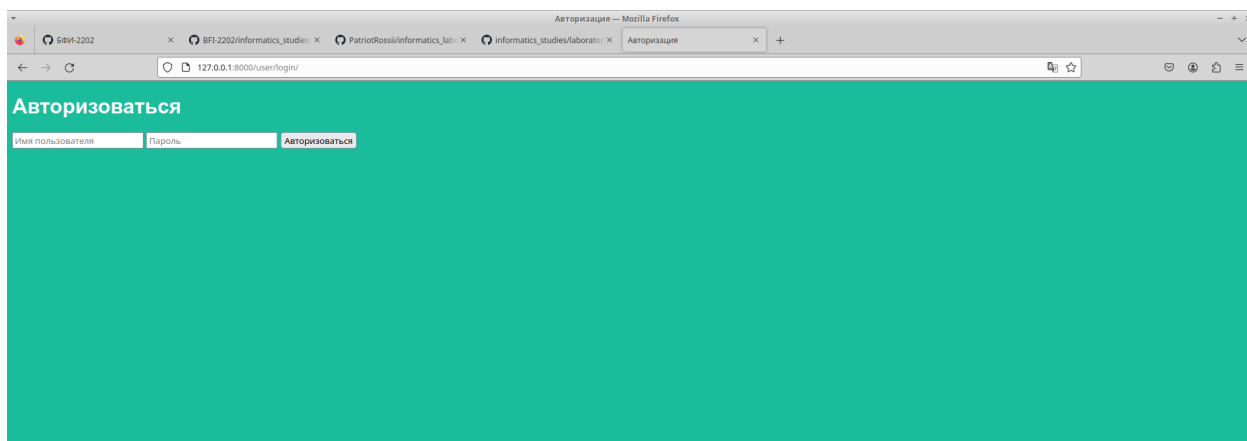
На рисунке ниже показан вид главной страницы для неавторизованного пользователя



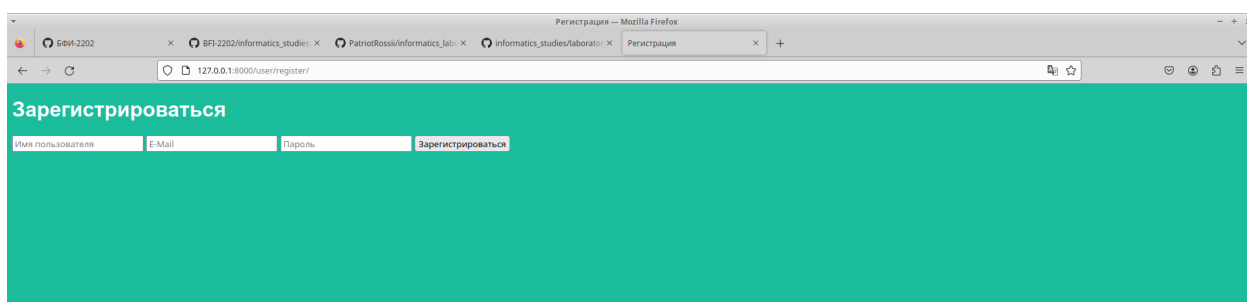
На рисунке ниже показан вид страницы отдельной статьи для неавторизованного пользователя



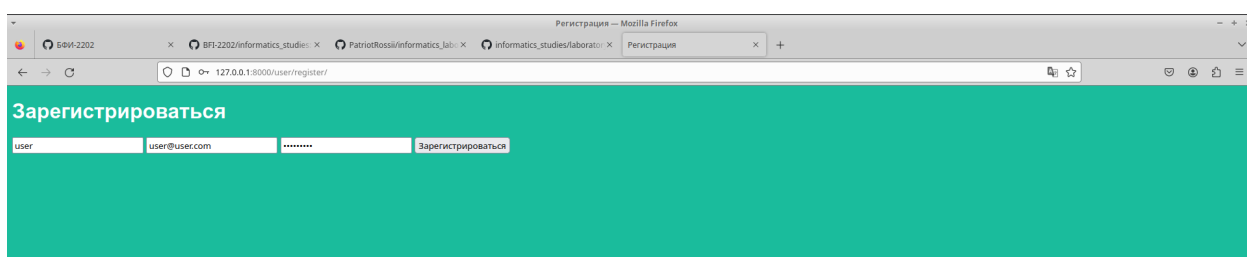
На рисунке ниже показан вид страницы авторизации для неавторизованного пользователя



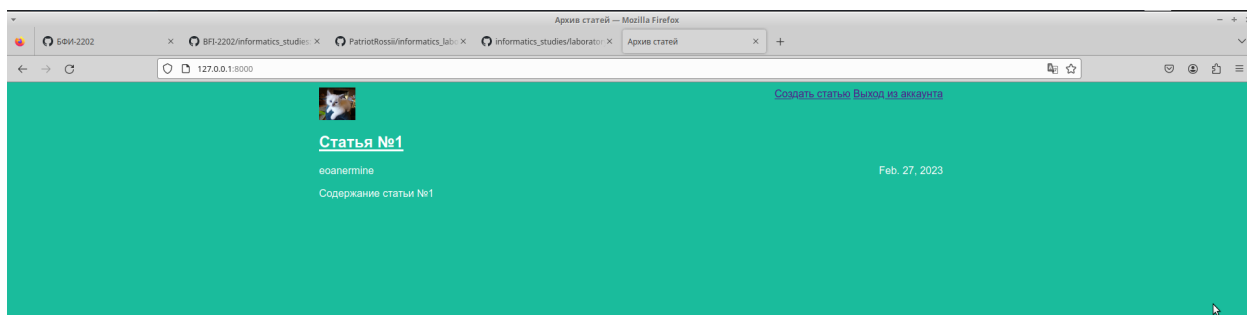
На рисунке ниже показан вид страницы регистрации для неавторизованного пользователя



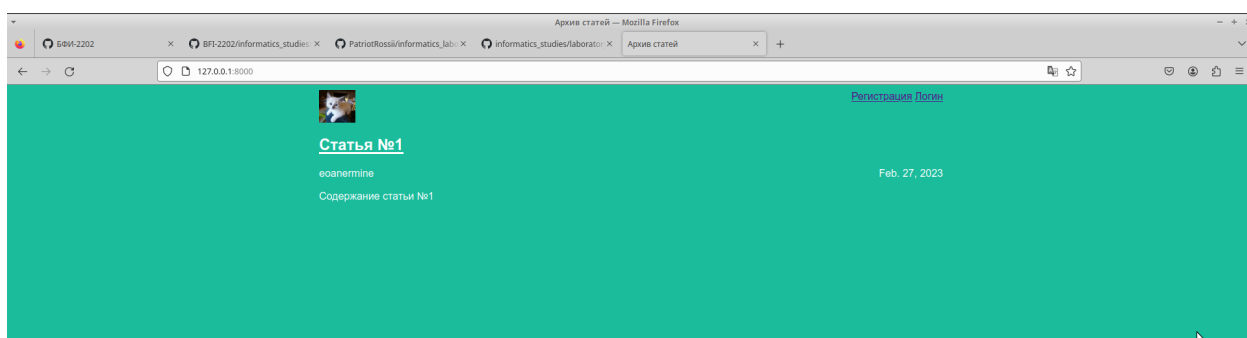
На рисунке ниже приведен пример заполнения формы регистрации



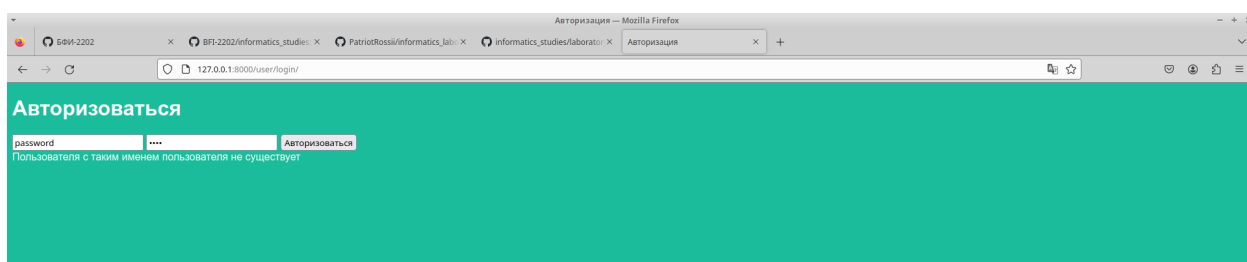
На рисунке ниже приведен результат нажатия кнопки «зарегистрироваться» при заполненной форме регистрации

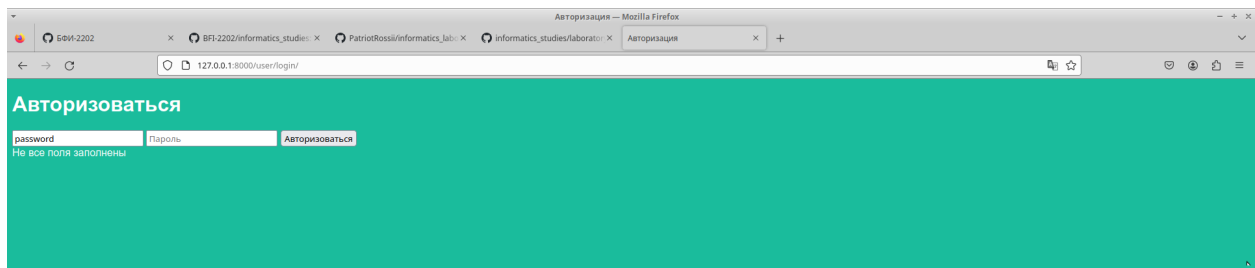


На рисунке ниже приведен результат нажатия кнопки «выход из аккаунта» для авторизованного пользователя

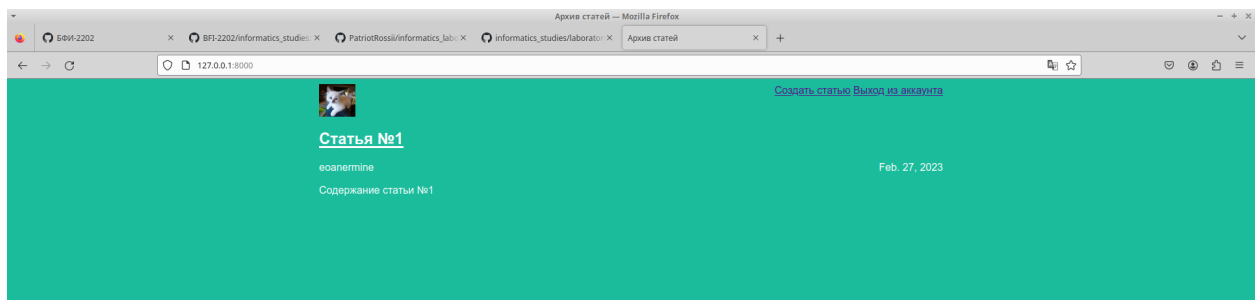


На рисунках ниже приведен результат нажатия кнопки «авторизоваться» при некорректных входных данных





На рисунке ниже приведен результат успешной авторизации



На рисунках ниже приведен результат нажатия кнопки «зарегистрироваться» при некорректных входных данных

