

**КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ
Ордена трудового Красного Знамени
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский технический университет связи и информатики»**

Кафедра «Электроника»



**Задание по дисциплине
«Прикладные информационные системы»**

Лабораторная работа № 3

Группа: УБСТ2201
ФИО: Лысов Вячеслав Константинович

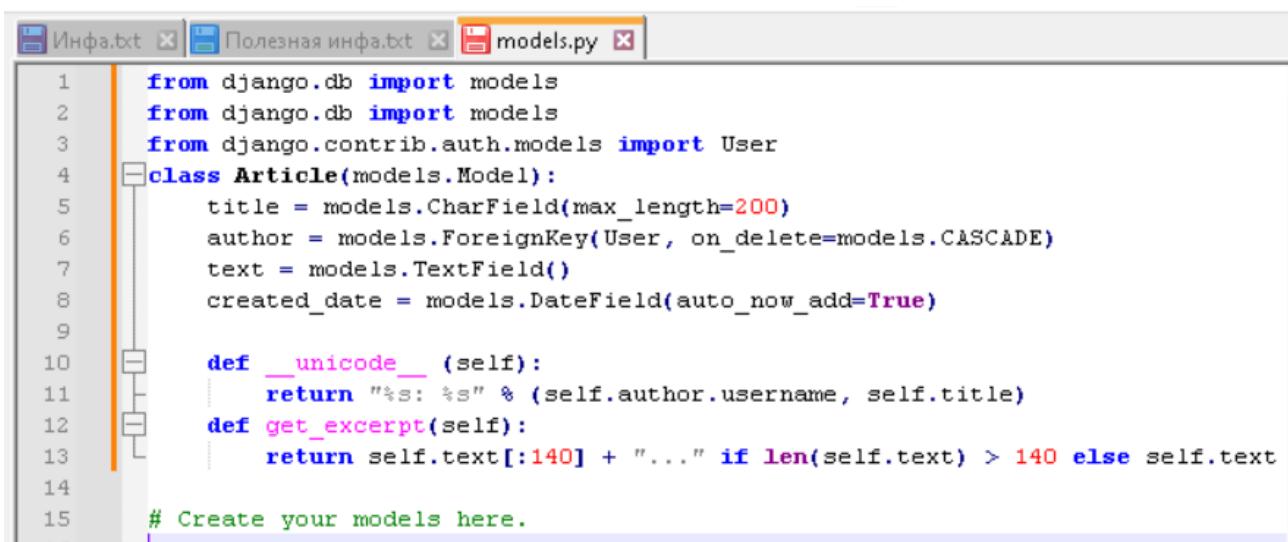
Москва 2025

Создал первую модель данных и ее регистрация в административном приложении Django.

```
D:\PIS\lab3>django-admin startproject blog  
D:\PIS\lab3>cd blog  
D:\PIS\lab3\blog>python manage.py startapp articles  
D:\PIS\lab3\blog>
```

Рисунок 1 — Создание первой модели данных.

В директории **articles** и в файл **models.py** сохраните код из лабораторной работы.



```
1 from django.db import models  
2 from django.contrib.auth.models import User  
3  
4 class Article(models.Model):  
5     title = models.CharField(max_length=200)  
6     author = models.ForeignKey(User, on_delete=models.CASCADE)  
7     text = models.TextField()  
8     created_date = models.DateTimeField(auto_now_add=True)  
9  
10    def __unicode__(self):  
11        return "%s: %s" % (self.author.username, self.title)  
12    def get_excerpt(self):  
13        return self.text[:140] + "..." if len(self.text) > 140 else self.text  
14  
15    # Create your models here.
```

Рисунок 2 — Создание файла models.py.

В этой же директории открыл файл **admin.py** (он ответственен за настройку страницы записей в административном приложении).



```
1 from django.contrib import admin  
2 from .models import Article  
3  
4 class ArticleAdmin(admin.ModelAdmin):  
5     list_display = ('title', 'author', 'get_excerpt')  
6     admin.site.register(Article, ArticleAdmin)  
7  
8     # Register your models here.
```

Рисунок 3 — Изменения в файле admin.py.

```
D:\PIS\lab3\blog>python manage.py makemigrations
Migrations for 'articles':
  articles\migrations\0001_initial.py
    + Create model Article

D:\PIS\lab3\blog>python manage.py migrate
Operations to perform:
  Apply all migrations: admin, articles, auth, contenttypes, sessions
Running migrations:
  Applying contenttypes.0001_initial... OK
  Applying auth.0001_initial... OK
  Applying admin.0001_initial... OK
  Applying admin.0002_logentry_remove_auto_add... OK
  Applying admin.0003_logentry_add_action_flag_choices... OK
  Applying articles.0001_initial... OK
  Applying contenttypes.0002_remove_content_type_name... OK
  Applying auth.0002_alter_permission_name_max_length... OK
  Applying auth.0003_alter_user_email_max_length... OK
  Applying auth.0004_alter_user_username_opts... OK
  Applying auth.0005_alter_user_last_login_null... OK
  Applying auth.0006_require_contenttypes_0002... OK
  Applying auth.0007_alter_validators_add_error_messages... OK
  Applying auth.0008_alter_user_username_max_length... OK
  Applying auth.0009_alter_user_last_name_max_length... OK
  Applying auth.0010_alter_group_name_max_length... OK
  Applying auth.0011_update_proxy_permissions... OK
  Applying auth.0012_alter_user_first_name_max_length... OK
  Applying sessions.0001_initial... OK
```

Рисунок 4 - Создание таблицы базы данных.

```
D:\PIS\lab3\blog>python manage.py createsuperuser
Username (leave blank to use 'lysov.vk'): lysov.vk
Email address: v.k.lysov@edu.mtuci.ru
Password:
Password (again):
This password is too short. It must contain at least 8 characters.
This password is too common.
Bypass password validation and create user anyway? [y/N]: y
Superuser created successfully.
```

Рисунок 5 - Создание супер пользователя.

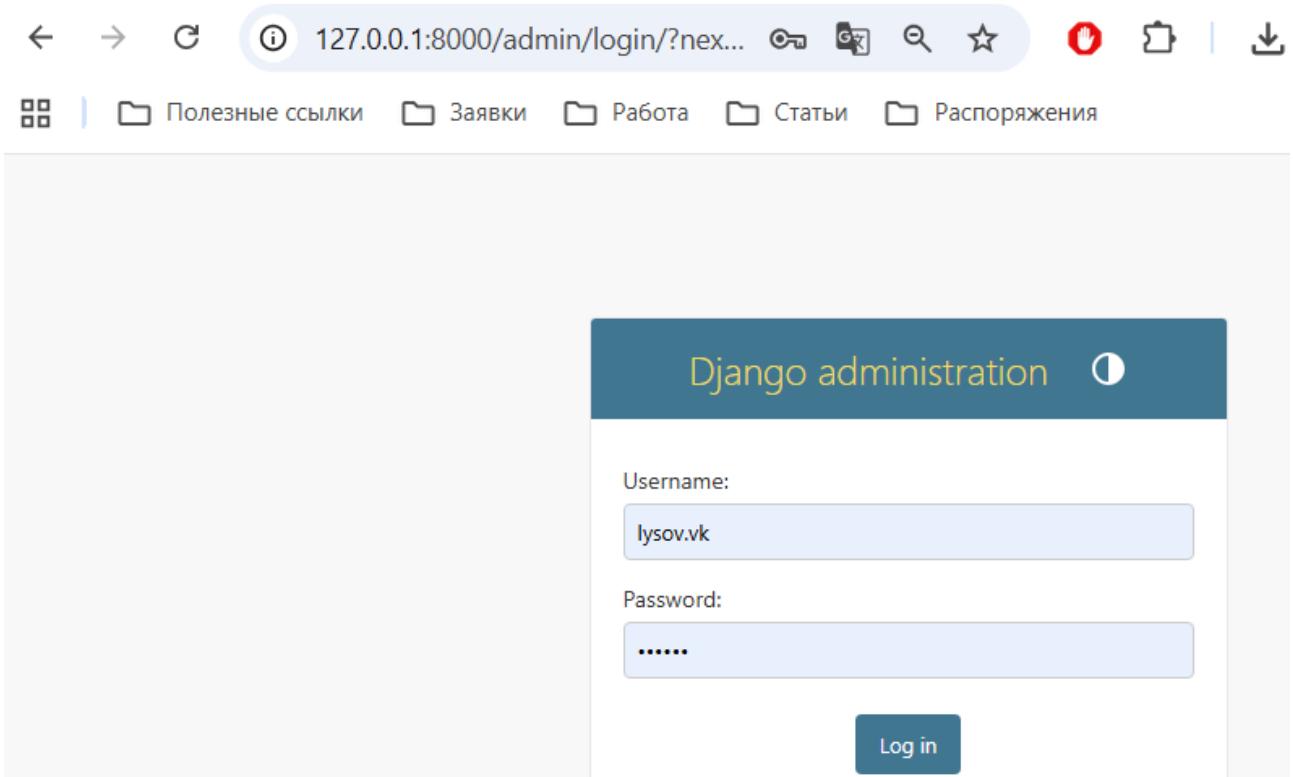


Рисунок 5 — Проверка работы сервера Django.

A screenshot of a web browser window. The address bar shows the URL "127.0.0.1:8000/ad...". The page title is "Django administration". The header includes "WELCOME, LYSOV.VK.", links for "VIEW SITE / CHANGE PASSWORD / LOG OUT", and a user icon. Below the header, the text "Site administration" is displayed. Under "ARTICLES", there is a table with a single row labeled "Articles" with "Add" and "Change" buttons. Under "AUTHENTICATION AND AUTHORIZATION", there are two tables: one for "Groups" with "Add" and "Change" buttons, and another for "Users" with "Add" and "Change" buttons.

Рисунок 6 — Успешная авторизация.

Задание:

- Перейдите во вкладку Articles и создайте 3 статьи, заполнив все поля.
- С помощью программы управления базами данных sqlite3 (например, SQLite Manager) откройте файл вашей базы данных текущего проекта, который хранится в папке проекта с именем, объявленным в настройках проектах в переменной «**DATABASES.NAME**». Найдите созданные в предыдущем пункте задания экземпляры записей. Измените текст одной записи и название статьи для другой. Создайте еще одну статью.

The screenshot shows a web-based administration interface for managing articles. On the left, there's a sidebar with a search bar and two main sections: 'ARTICLES' (with a 'Articles' link and a '+ Add' button) and 'AUTHENTICATION AND AUTHORIZATION' (with 'Groups' and 'Users' links, each also having a '+ Add' button). The main content area has a green header bar stating 'The article "Article object (3)" was added successfully.' Below this, a table lists three articles with columns for TITLE, AUTHOR, GET EXCERPT, and CREATED DATE. The articles are titled 'Test3', 'Test2', and 'Test1', all authored by 'lysov.vk'. At the bottom of the table, it says '3 articles'. To the right of the table is a large 'ADD ARTICLE' button with a '+' icon.

Action:	-----	Go	0 of 3 selected	
<input type="checkbox"/>	TITLE	AUTHOR	GET EXCERPT	CREATED DATE
<input type="checkbox"/>	Test3	lysov.vk	Test3	Nov. 20, 2025
<input type="checkbox"/>	Test2	lysov.vk	Test2	Nov. 20, 2025
<input type="checkbox"/>	Test1	lysov.vk	Test1	Nov. 20, 2025

Рисунок 7 — Создание трёх статей.

The screenshot shows two database queries in SQLite Manager:

```
DB 1 #1
1 SELECT name FROM sqlite_master
2 WHERE type IN ('table','view') AND name NOT LIKE 'sqlite_%' ORDER BY 1
```

Result of Query #1:

#	name
1	articles_article
2	auth_group
3	auth_group_permissions
4	auth_permission
5	auth_user
6	auth_user_groups
7	auth_user_user_permissions
8	django_admin_log
9	django_content_type
10	django_migrations

10 First Prev Next Last


```
DB 1 #2
1 SELECT * from "articles_article" LIMIT 10
```

Result of Query #2:

#	id	title	text	created_date	author_id
1	1	Test1	Test1	2025-11-20	1
2	2	Test2	Test2	2025-11-20	1
3	3	Test3	Test3	2025-11-20	1

Export Copy Execute (Enter) Close

Рисунок 8 — Отображение базы через SQLite Manager.

The screenshot shows a list of articles with updated titles and texts:

<input type="checkbox"/>	TITLE	AUTHOR	GET EXCERPT	CREATED DATE
<input type="checkbox"/>	NEWTest4	lysov.vk	NEWTest4	Nov. 20, 2025
<input type="checkbox"/>	Test3	lysov.vk	Test3	Nov. 20, 2025
<input type="checkbox"/>	ОБНОВЛЕННЫЙ ЗАГОЛОВОК: Test2	lysov.vk	Test2	Nov. 20, 2025
<input type="checkbox"/>	Test1	lysov.vk	ОБНОВЛЕННЫЙ ТЕКСТ: Test1	Nov. 20, 2025

4 articles

Рисунок 9 — Обновленные данные.

Динамическое генерирование шаблона для вывода всех экземпляров этой модели

```
lab3 > blog > articles > templates > 5 archive.html > 6 html
1   <!DOCTYPE html>
2   <html>
3     <head>
4       <title>Архив статей</title>
5       {% load static %}
6     </head>
7     <body>
8       <div class="header">
9         
10      </div>
11      <div class="archive">
12        {% for post in posts %}
13          <div class="one-post">
14            <h2 class="post-title">{{ post.title }}</h2>
15            <div class="article-info">
16              <div class="article-author">{{ post.author.username }}</div>
17              <div class="article-created-date">{{ post.created_date }}</div>
18            </div>
19            <p class="article-text">{{ post.get_excerpt }}</p>
20          </div>
21        {% endfor %}
22      </div>
23    </body>
24  </html>
```

Рисунок 10 — Создание файла archive.html.

```
lab3 > blog > articles > 7 views.py > ...
1   from django.shortcuts import render
2   from .models import Article
3
4   def archive(request):
5     return render(request, 'archive.html', {"posts": Article.objects.all()})
6
7   # Create your views here.
8   |
```

Рисунок 10 — Создание представления в файле views.py.

```
from django.contrib import admin
from django.urls import path, include
from articles import views

urlpatterns = [
    path('admin/', admin.site.urls),
    path('', views.archive, name='archive'),
]
```

Рисунок 11 — Создание представления в файле urls.py.

```
STATIC_URL = 'static/'  
STATICFILES_DIRS = [  
    os.path.join(BASE_DIR, 'articles/static'),  
]
```

Рисунок 12 — Создание представления в файле settings.py.



Test1

lysov.vk
Nov. 20, 2025

ОБНОВЛЕННЫЙ ТЕКСТ: Test1

ОБНОВЛЕННЫЙ ЗАГОЛОВОК: Test2

lysov.vk
Nov. 20, 2025

Test2

Test3

lysov.vk
Nov. 20, 2025

Test3

NEWTest4

lysov.vk
Nov. 20, 2025

NEWTest4

Рисунок 13 — Главная страница.

Задание:

- Откройте файл базы данных, где хранятся экземпляры статей текущего проекта, с помощью программы управления базами данных sqlite3 и добавьте новую запись в блог через менеджера базы;
- Загрузите ваш проект на любой гит-репозиторий (GitHub, GitLab, Google Code, Bitbucket и т.п.).

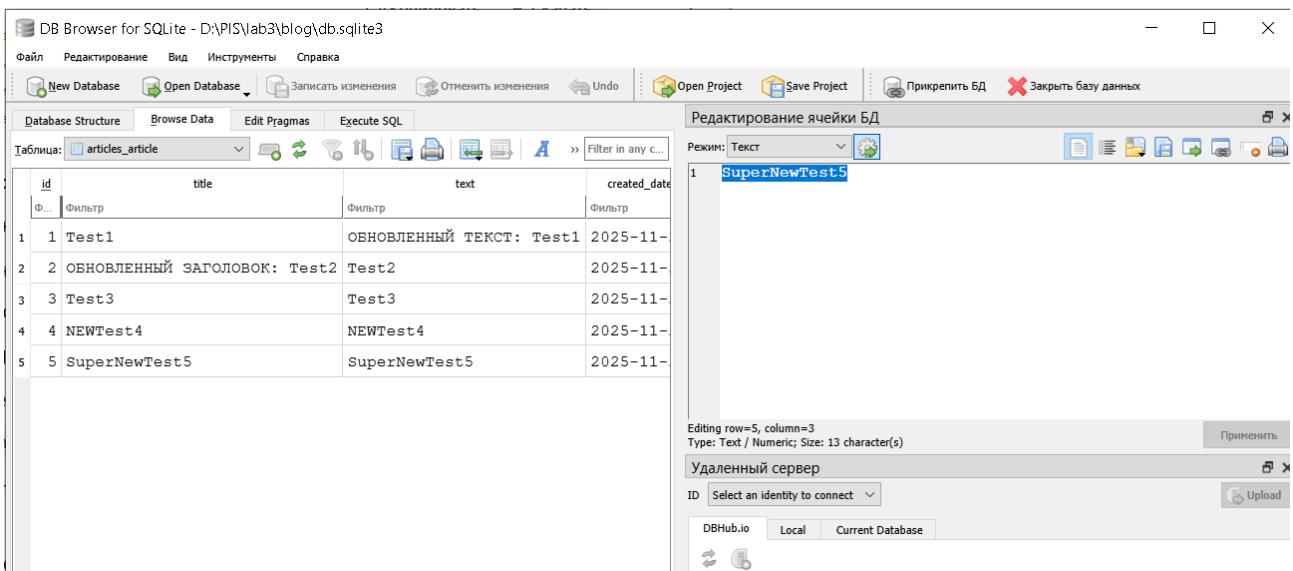


Рисунок 14 — Изменение и добавление данных через программу DB Browser for SQLite.



Test1

lysov.vk
Nov. 20, 2025

ОБНОВЛЕННЫЙ ТЕКСТ: Test1

ОБНОВЛЕННЫЙ ЗАГОЛОВОК: Test2

lysov.vk
Nov. 20, 2025

Test2

Test3

lysov.vk
Nov. 20, 2025

Test3

NEWTest4

lysov.vk
Nov. 20, 2025

NEWTest4

SuperNewTest5

lysov.vk
Nov. 20, 2025

SuperNewTest5

Рисунок 14 — Проверка изменений на главной странице.