

**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 «Сетевые информационные технологии и сервисы»



Отчет по лабораторной работе

по дисциплине

«Прикладные информационные системы»

на тему:

**«Создание первой модели данных и ее регистрация в административном
приложении Django»**

Выполнил студент 4 курса

Группы: УБСТ 2203

Музыченко Г. Н.

Москва, 2025

Ход работы

1. Создание проекта

Создадим директорию проекта, а затем приложение *articles* (рис. 1)

```
C:\Users\Public\Documents\lab3\blog>python manage.py startapp articles_
```

Рисунок 1 – Вывод командной строки

Добавляем код в models.py

```
1  django.db import models
2  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.DateField(auto_now_add=True)
9
10     def __str__(self):
11         return "%s: %s" % (self.author.username, self.title)
12
13     def get_excerpt(self):
14         return self.text[:140] + "..." if len(self.text) > 140 else self.text
```

Рисунок 2 – Содержимое models.py

Добавляем код в admin.py

```
C: > Users > Public > Documents > lab3 > blog > articles > admin.py > ...
1  from django.contrib import admin
2  from .models import Article
3
4  class ArticleAdmin(admin.ModelAdmin):
5      list_display = ('title', 'author', 'get_excerpt', 'created_date')
6      ⚡
7  admin.site.register(Article, ArticleAdmin)
```

Рисунок 3 – Содержимое admin.py

```

INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'articles'
]

```

Рисунок 4 – Содержимое блока INSTALLED_APPS

```

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': os.path.join(BASE_DIR, 'db_blog.sqlite3')
    }
}

```

Рисунок 5 – Содержимое блока DATABASES

Создаем таблицы БД на основе Article

```

C:\Users\Public\Documents\lab3\blog>python manage.py migrate
Operations to perform:
  Apply all migrations: admin, 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 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 6 – Вывод командной строки

Создаем суперюзера и запускаем сервер (рис. 7)

```
Superuser created successfully.

C:\Users\Public\Documents\lab3\blog>python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...
```

Рисунок 7 – Вывод командной строки

2. Выполнение заданий

Создаем три статьи (заголовок, автор, содержание).

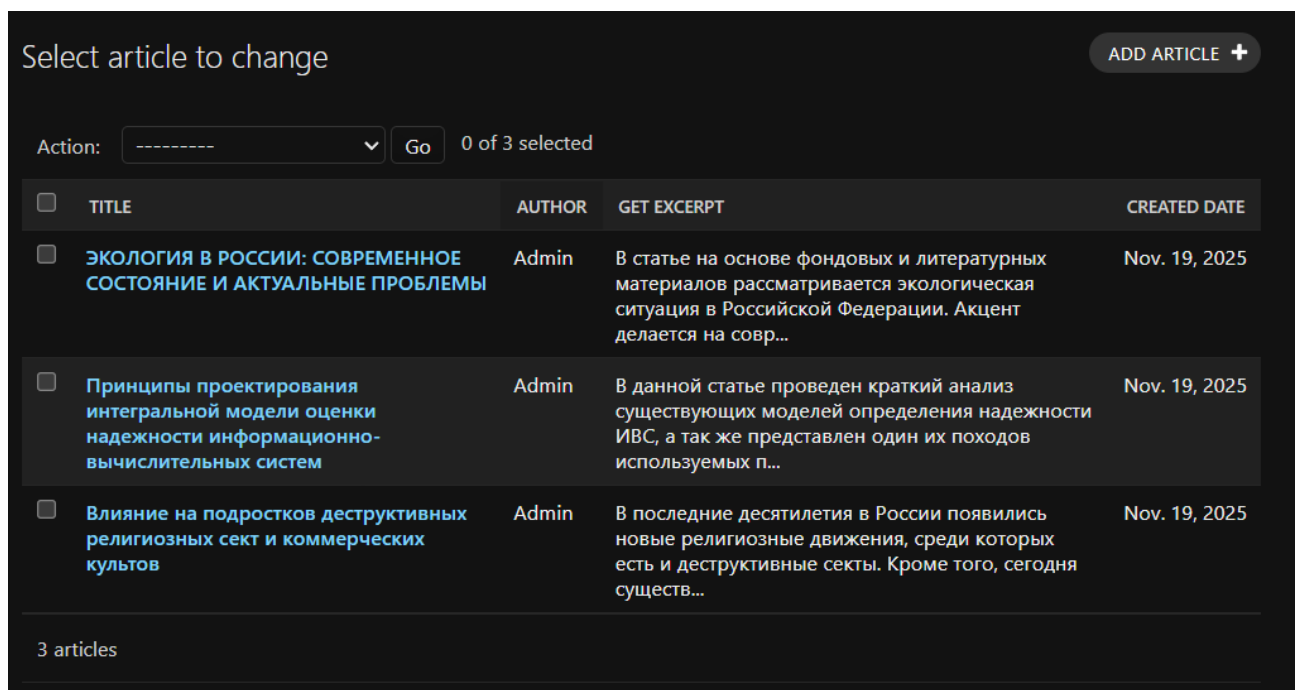


Рисунок 8 – Созданные статьи

Откроем файл базы данных с помощью DB Browser

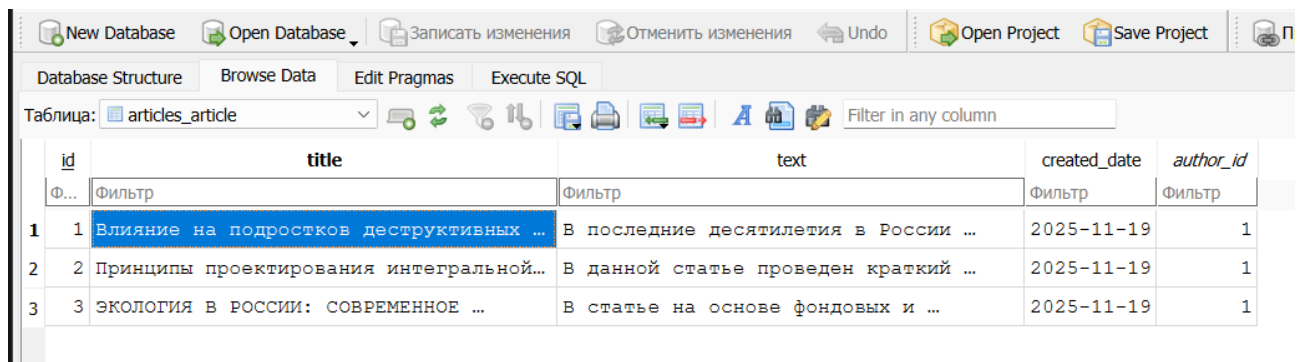


Рисунок 9 – Интерфейс программы

Добавляем статью с помощью DB Browser с помощью интерфейса программы, результат на рисунке ниже (рис. 10)

<input type="checkbox"/>	TITLE	AUTHOR	GET EXCERPT
<input type="checkbox"/>	Заголовок	Admin	Содержание
<input type="checkbox"/>	ЭКОЛОГИЯ В РОССИИ: СОВРЕМЕННОЕ СОСТОЯНИЕ И АКТУАЛЬНЫЕ ПРОБЛЕМЫ	Admin	В статье на основе фон... Акцент делается на сов...
<input type="checkbox"/>	Принципы проектирования интегральной модели оценки надежности информационно-вычислительных систем	Admin	В данной статье провед... походов используемых...
<input type="checkbox"/>	Влияние на подростков деструктивных религиозных сект	Admin	В последние десятилет... того, сегодня существ...

4 articles

Рисунок 10 – Интерфейс программы

Либо через SQL-запрос:

```
INSERT INTO articles_article (title, author_id, text, created_date)
VALUES ('Новая статья', 1, 'Содержание статьи бла-бла-бла', date('now'));
```

Создаем файл archive.html в директории templates

```
C: > Users > Public > Documents > lab3 > blog > articles > templates > archive.html > body
1 <body>
2   <div class="header">
3   </div>
4   <div class="archive">
5   </div>
6   </body>
```

Рисунок 11 – Содержание файла archive.html

Далее настроим urls.py чтобы главная страница вызывала views.archive, а `127.0.0.1:8000/admin` – в интерфейс администратора (рис. 12).

```
C: > Users > Public > Documents > lab3 > blog > blog > 📄 urls.py > ...
1  from django.contrib import admin
2  from django.urls import path
3  from articles import views
4
5  urlpatterns = [
6      path('admin/', admin.site.urls),
7      📌 path('', views.archive, name='archive'),
8  ]
```

Рисунок 12 – Содержание файла archive.html

Если применить style.css и добавить его в archive.html, то главная страница будет выглядеть следующим образом (рис. 13)

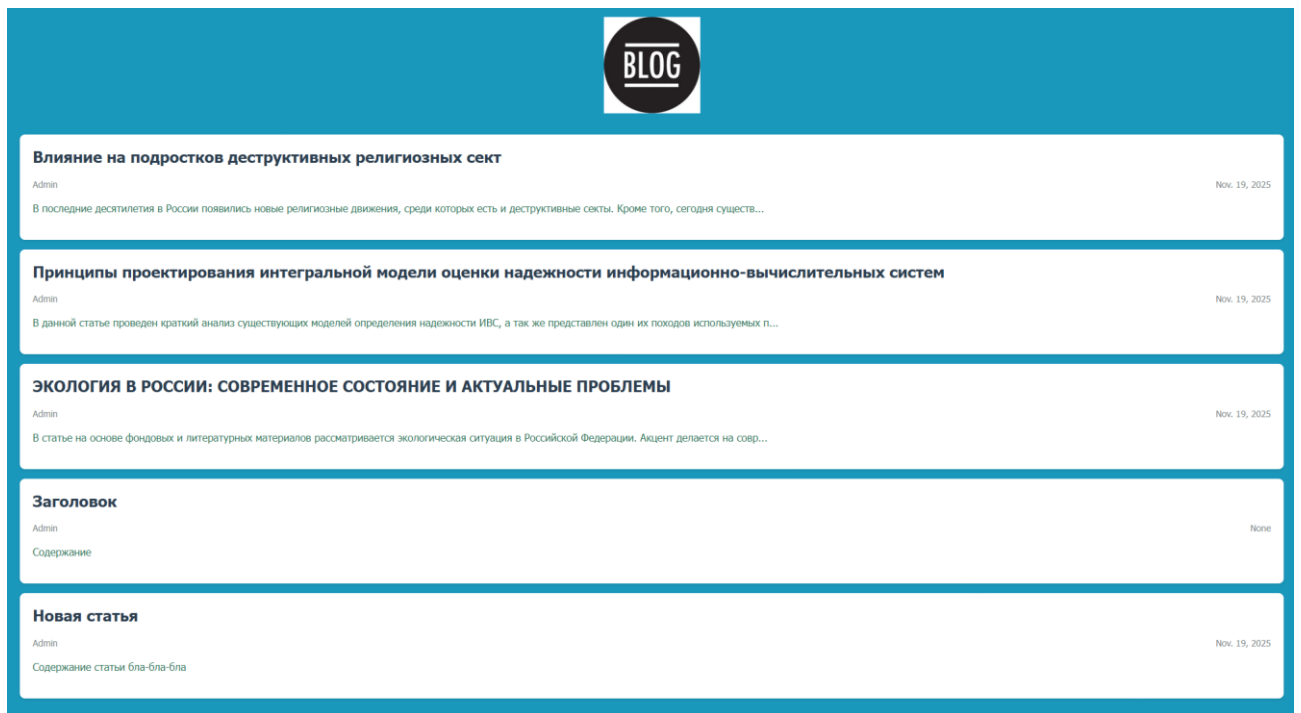


Рисунок 13 – Главная страница