**МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ РФ**

**Ордена Трудового Красного Знамени федеральное государственное бюджетное образовательное учреждение высшего образования «Московский технический университет связи и информатики»**

Кафедра математической кибернетики и информационных технологий

Лабораторная работа №3 по дисциплине «Web-программирование»:

«Создание модели данных и статей в Django»

Выполнил студент

группы БФИ1901

Гусев Никита

Проверила:

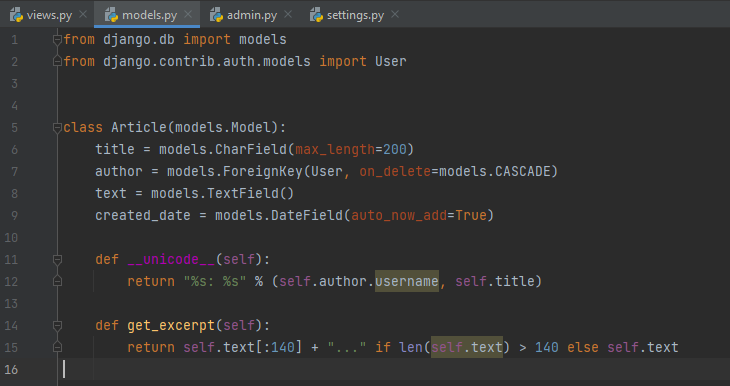
Полянцева Ксения Андреевна

Москва 2021 г.

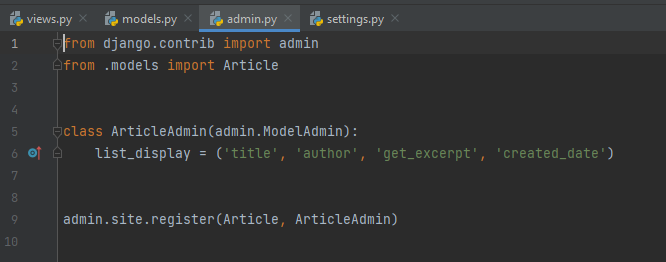
# Ход работы

## Задание №1. Создание первой модели данных и ее регистрация в административном приложении Django

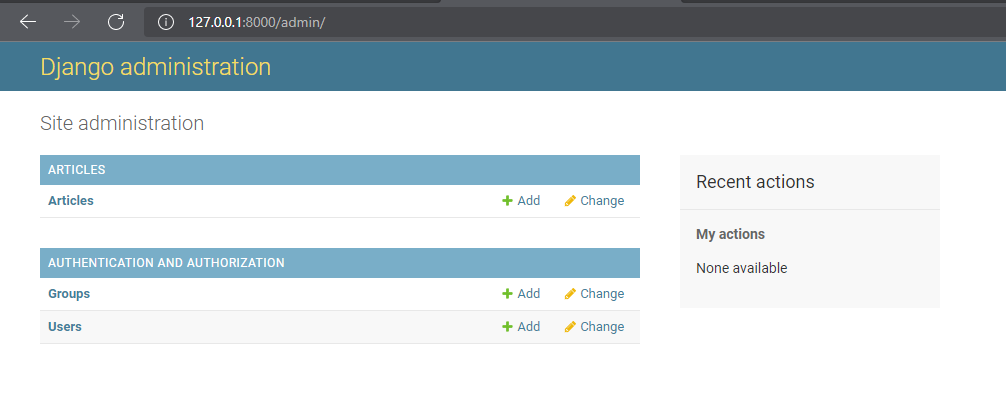
С помощью команды **django-admin.py startproject blog** создадим проект с названием blog в директории lab3. Затем, перейдем в папку blog и выполним команду **python manage.py startapp articles**. Перейдём в директорию articles и в файл models.py сохраним код. Файл models.py показан на рисунке 1:

  
Рисунок 1 – Файл models.py

В этой же директории отредактируем файл admin.py и сохраним в нем код. Файл admin.py показан на рисунке 2:

  
Рисунок 2 – Файл admin.py

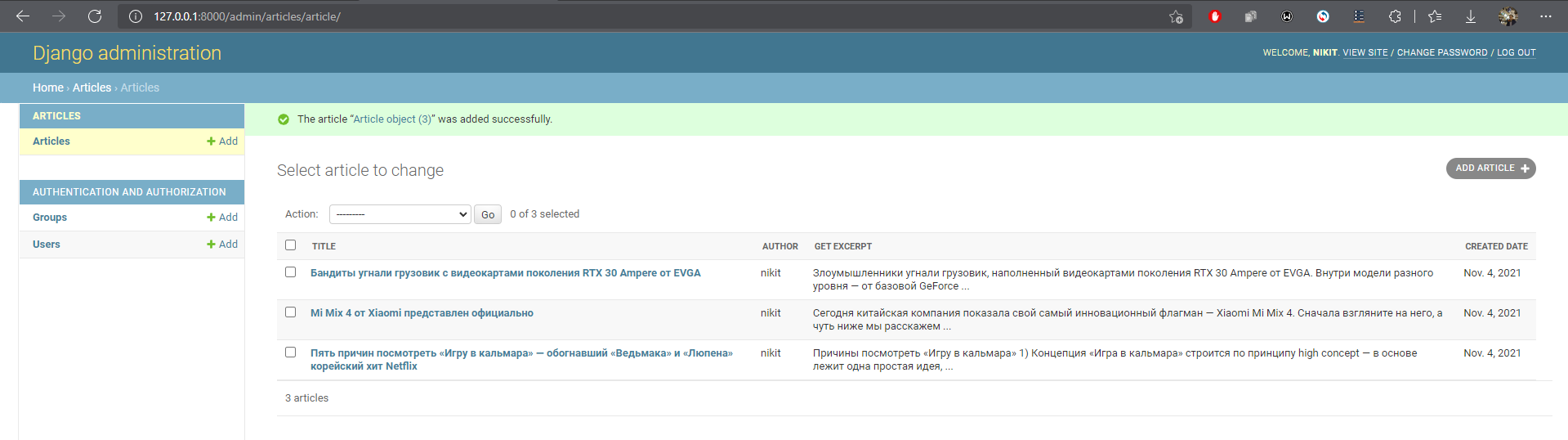
Чтобы убедиться в правильности добавления данных, откроем административную панель по адресу: <http://127.0.0.1:8000/admin/>. Она показана на рисунке 3:

  
Рисунок 3 – Административная панель

**Задание:**

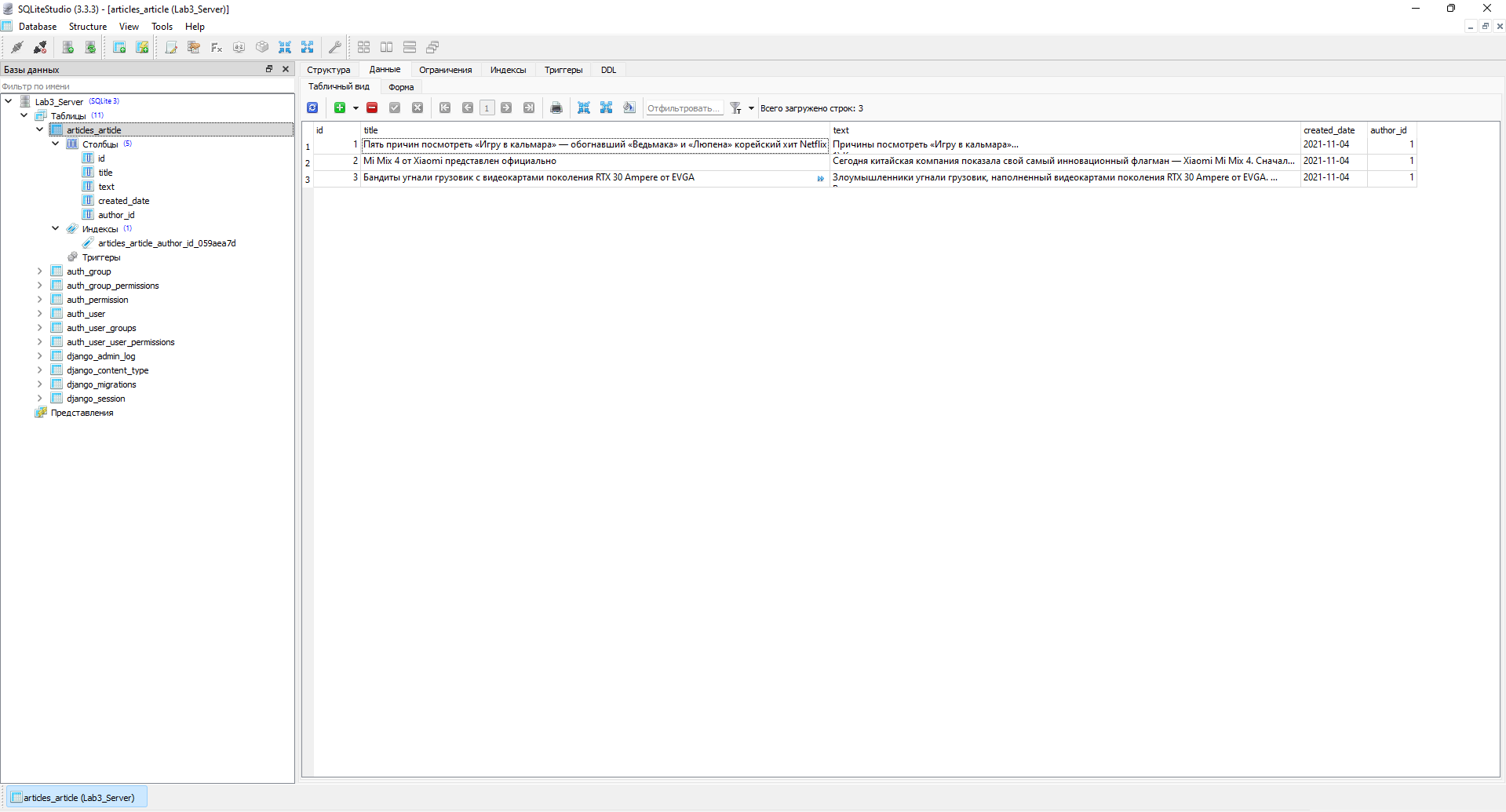
* Перейдите во вкладку Articles и создайте 3 статьи, заполнив все поля.

Результат добавления статей показан на рисунке 4:

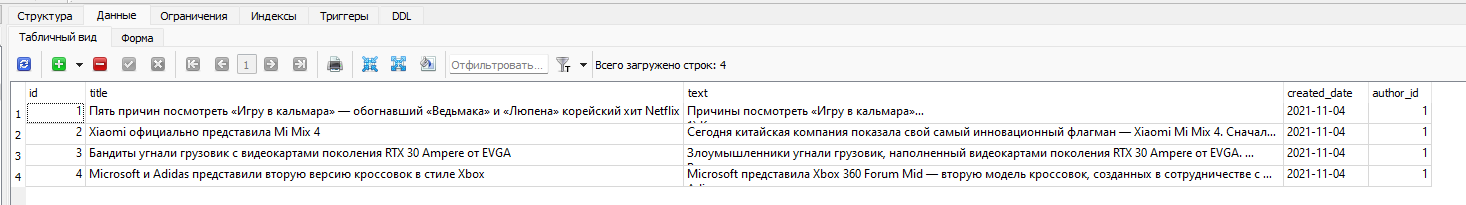
  
Рисунок 4 –Добавленные статьи

* 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.

Откроем программу SQLiteStudio, в ней откроем файл базы данных нашего проекта и найдём добавленные статьи. Они показаны на рисунке 5:

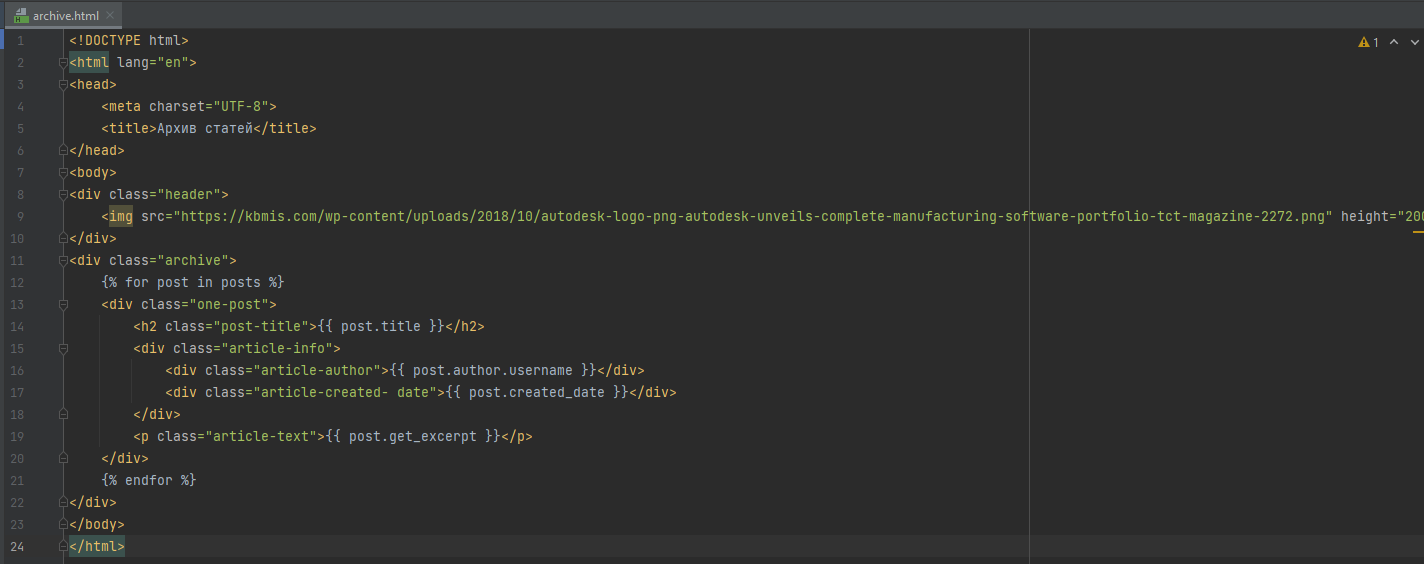
  
Рисунок 5 – Статьи в базе данных

Согласно заданию, отредактируем заголовок одной из статей и добавим ещё одну статью. Результат выполнения показан на рисунке 6:

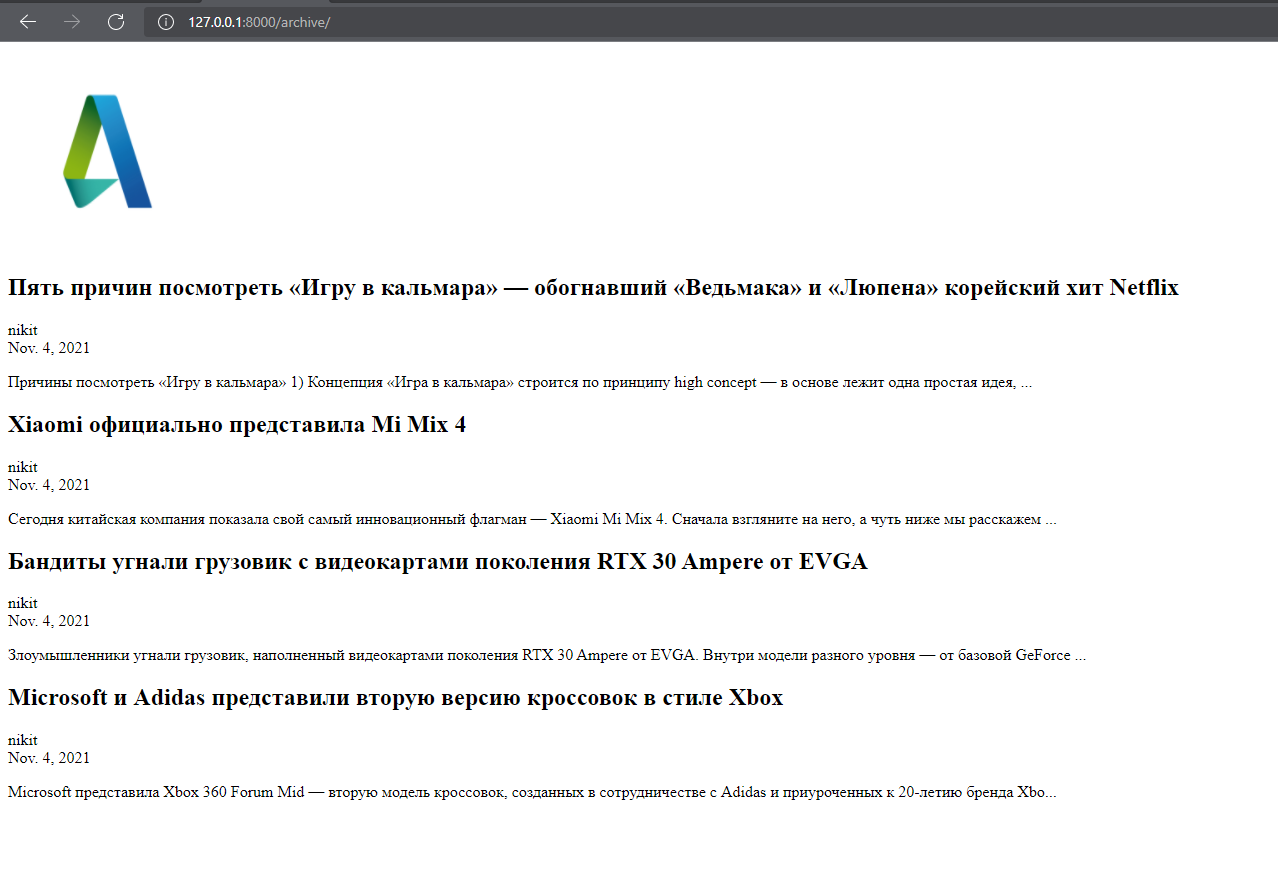
  
Рисунок 6 – Внесение изменений

## Задание №2. Динамическое генерирование шаблона для вывода всех экземпляров этой модели

В директории articles создадим папку templates, внутри которой создадим файл archive.html. В созданном файле шаблона в качестве названия страницы указала фразу «Архив статей». Затем в тег <body> добавим 2 тега <div>. Во внутрь тега div с классом header добавим изображение. Во внутрь тега div с классом archive добавим шаблон для отображения одной статьи. Для того, чтобы разметка поддерживала отображение сразу многих записей, добавим строку начала цикла и строку его завершения. Результат выполнения вышеописанных действий показан на рисунке 7:

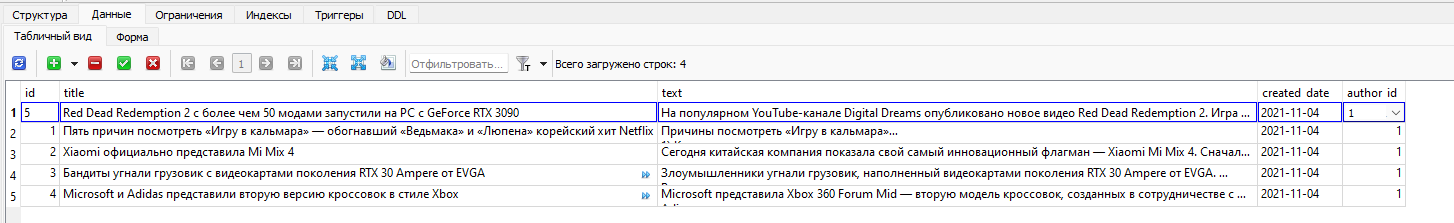
  
Рисунок 7 – Файл archive.html

В файле views.py в директории articles создадим представление archive, которое будет возвращать html-страницу со всеми созданными постами в текущем проекте. После выполнения всех действий, описанных выше должен получиться такой сайт, который показан на рисунке 8:

  
Рисунок 8 – Результат выполнения

* Откройте файл базы данных, где хранятся экземпляры статей текущего проекта, с помощью программы управления базами данных sqlite3 и добавьте новую запись в блог через менеджер базы.

Откроем программу SQLiteStudio, в ней откроем файл базы данных нашего проекта и найдём добавленные статьи. Добавим ещё одну статью, как показано на рисунке 9:

  
Рисунок 9 – Добавление статьи в менеджере

Результат показан на рисунке 10:

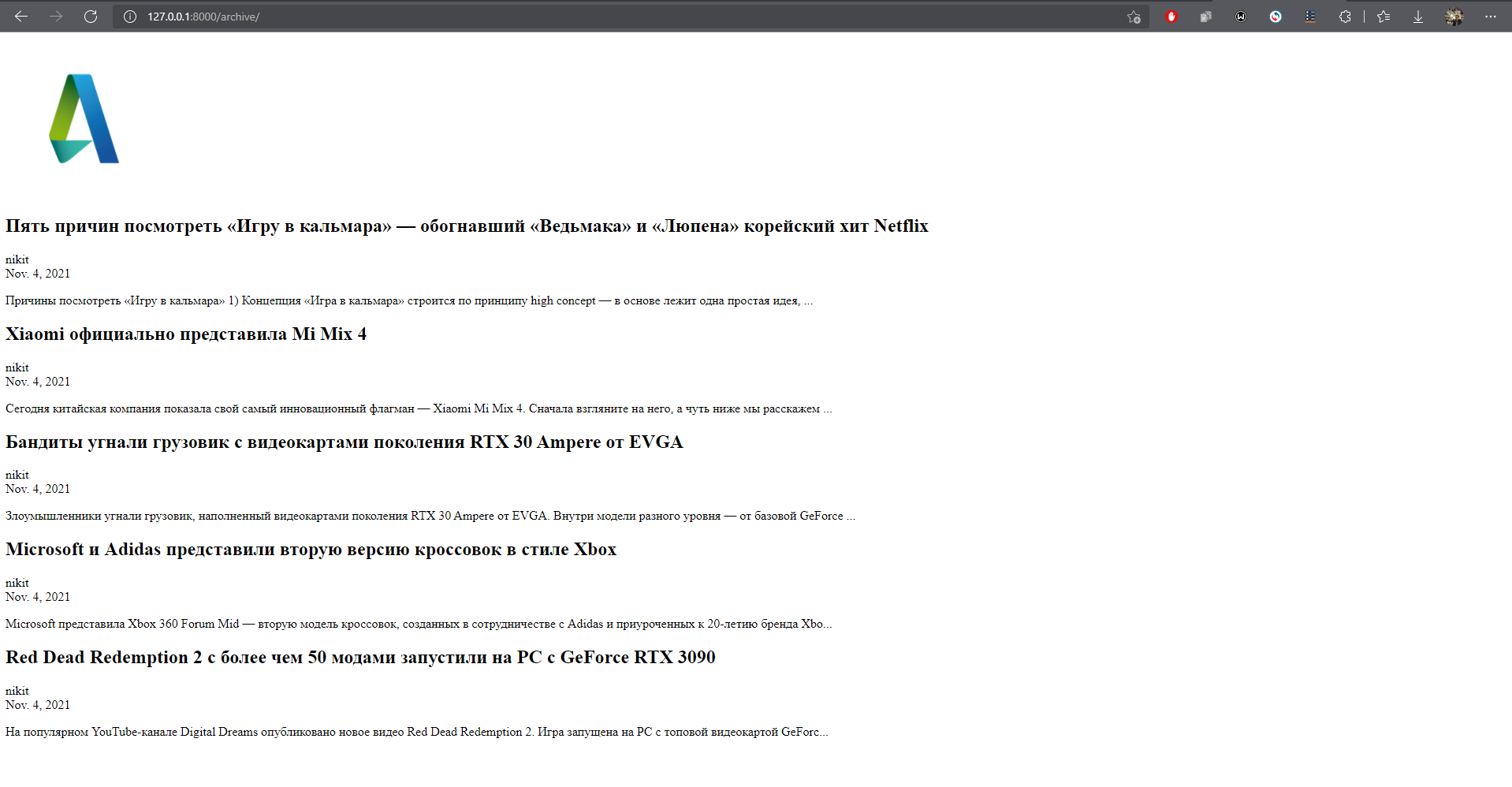


Рисунок 10 - Результат работы