**Федеральное агентство связи**

**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 №4**

по предмету «Прикладные информационные системы»

Выполнил: студент группы УБСТ 2304

Ивлев Константин Евгеньевич

Руководитель:

Кутейников Иван Алексеевич

Москва 2025

Создание проекта.

В директории lab4 был создан новый проект blog, произведена первичная настройка и перенесено приложение articles.

Рисунок 1. Добавление приложения в settings



Рисунок 1. Добавление приложения в settings

Для того, чтобы можно было бы прочитать значение идентификатора (id) отдельного поста, в файле urls.py было добавлено регулярное выражение. Рисунок 2 – редактирование urls.py. Добавление регулярного выражения

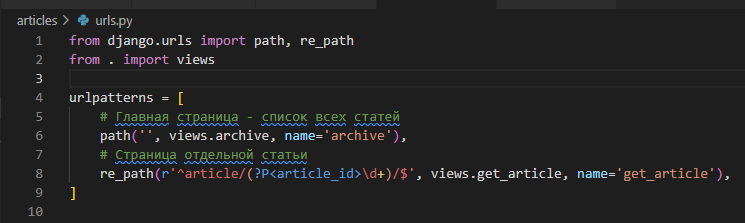


Рисунок 2 – редактирование urls.py. Добавление регулярного выражения

В файле views.py из директории articles добавлена функция get\_article. Рисунок 3 – редактирование файла views.py

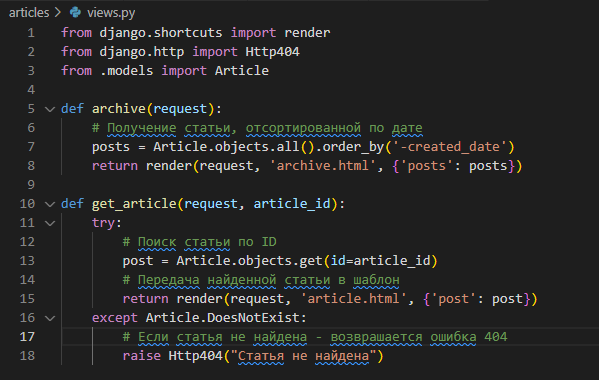


Рисунок 3 – редактирование файла views.py

Рисунок 4 – главная страница блога

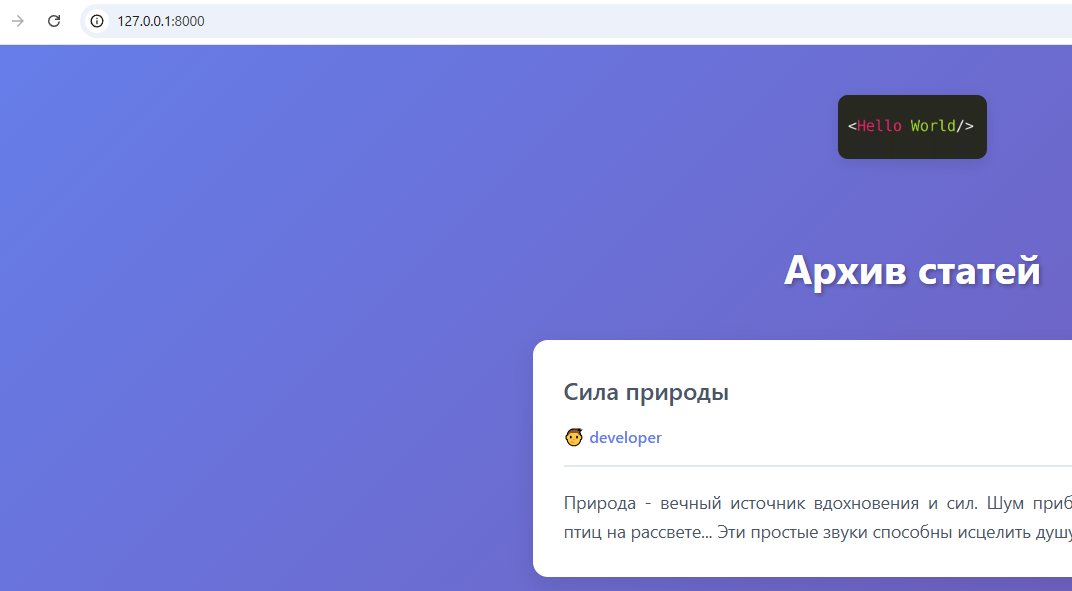


Рисунок 4 – главная страница блога

Рисунок 5 – переход на вторую статью

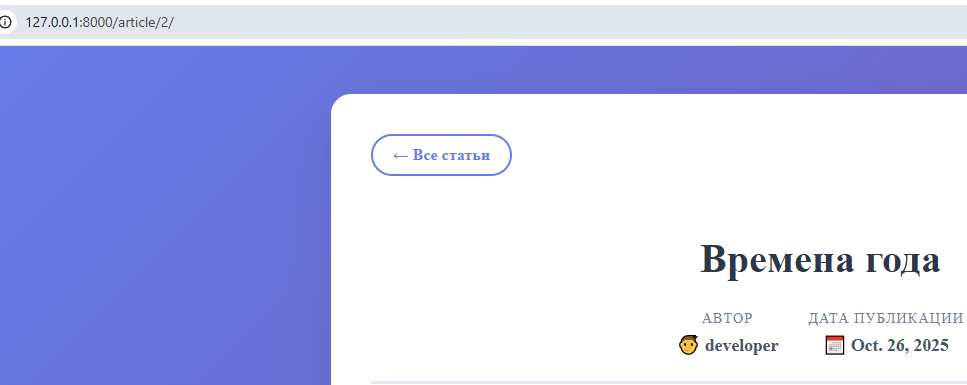


Рисунок 5 – переход на вторую статью

Рисунок 6 – переход на третью статью

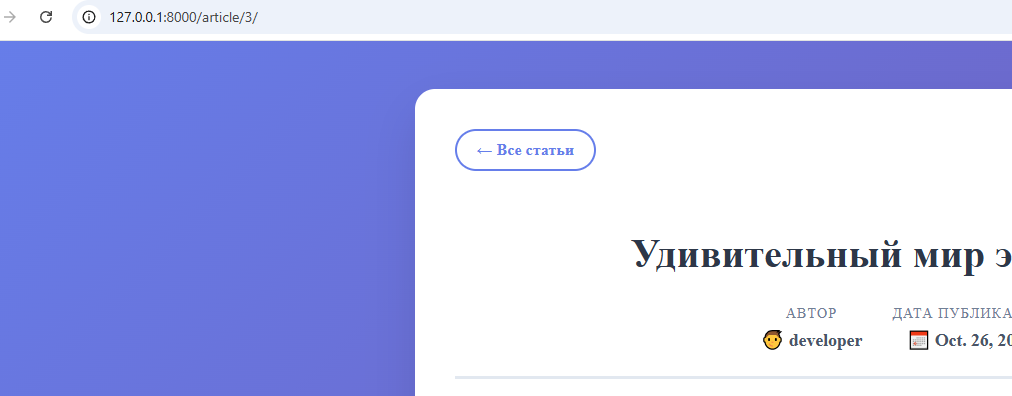


Рисунок 6 – переход на третью статью

Логика работы web-приложения:

* Пользователь заходит на главную страницу (/) - срабатывает views.archive
* Страница отображает список статей с заголовками-ссылками
* При клике на заголовок - открывается соответствующая статья по адресу /article/ID статьи/
* Django находит маршрут в urls.py с регулярным выражением
* Извлекает article\_id из URL и передает в views.get\_article
* Функция находит статью в базе данных и отображает полный текст