**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 №3**

по предмету «Прикладные информационные системы»

Выполнил: студент группы УБСТ 2304

Ивлев Константин Евгеньевич

Руководитель:

Кутейников Иван Алексеевич

Москва 2025

Создание проекта.

Новое Web-приложение **blog** было создано при помощи команд:

django-admin startproject blog

python manage.py startapp articles

рисунок 1. Создание проекта

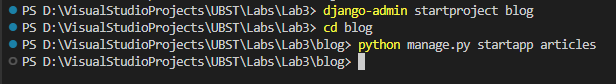


рисунок 1. Создание проекта

рисунок 2. Добавление приложения в файл settings

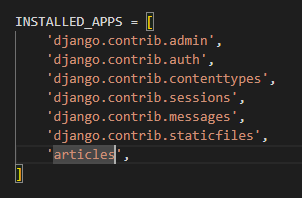


рисунок 2. Добавление приложения в файл settings

Рисунок 3. Редактирование файла models.py

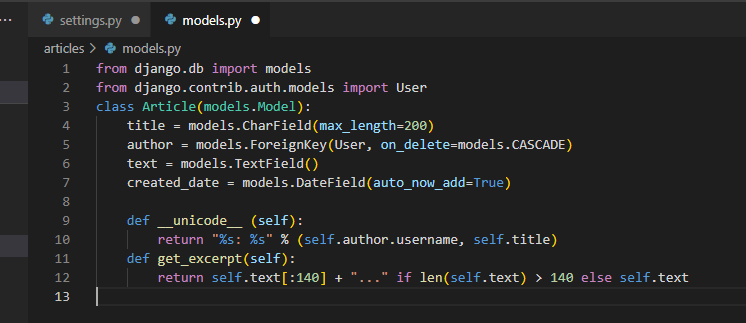


Рисунок 3. Редактирование файла models.py

Будущая модель статей будет иметь 4 поля: заголовок, автор, текст и время создания (в последнем значение будет устанавливаться автоматически).

Метод get\_excerpt позволяет в списке всех статей выводить текст статьи не целиком, а показывать первые 140 символов.

Рисунок 4. Редактирование файла admin.py



Рисунок 4. Редактирование файла admin.py

Класс ArticleAdmin нужен для того, чтобы, используя декларативный стиль, описать то, каким образом модель Article должна отображаться в административной панели.

Рисунок 5. Создание суперпользователя

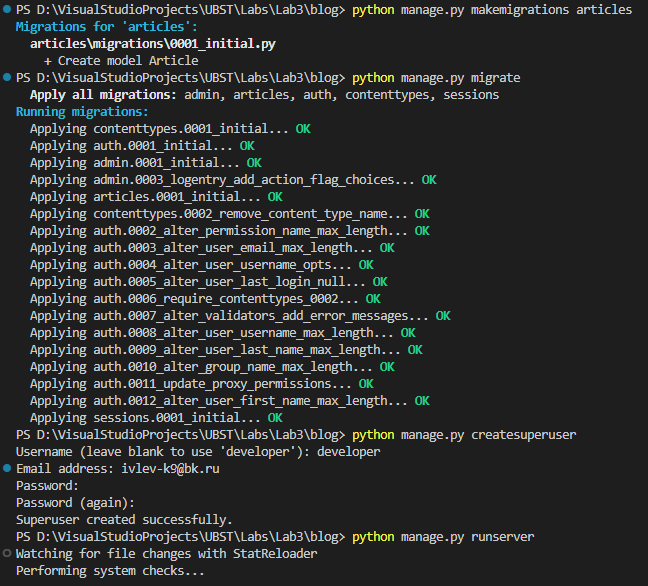


Рисунок 5. Создание суперпользователя

Рисунок 6. Запуск админ-панели

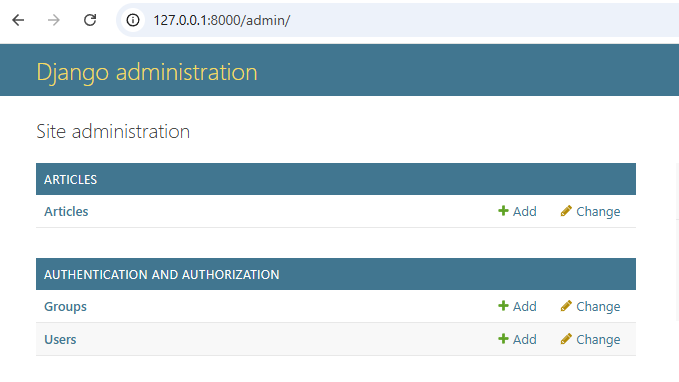


Рисунок 6. Запуск админ-панели

Рисунок 7. Добавление трёх статей

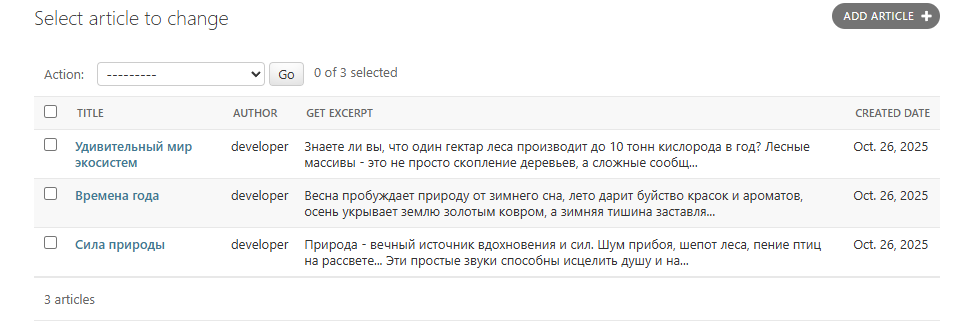


Рисунок 7. Добавление трёх статей

для отображения списка всех статей (архива) на главной странице блога нужен шаблон archive.html. Это основной шаблон, который отображается на главной странице сайта

Рисунок 8. Создание и редактирование шаблона archive.html

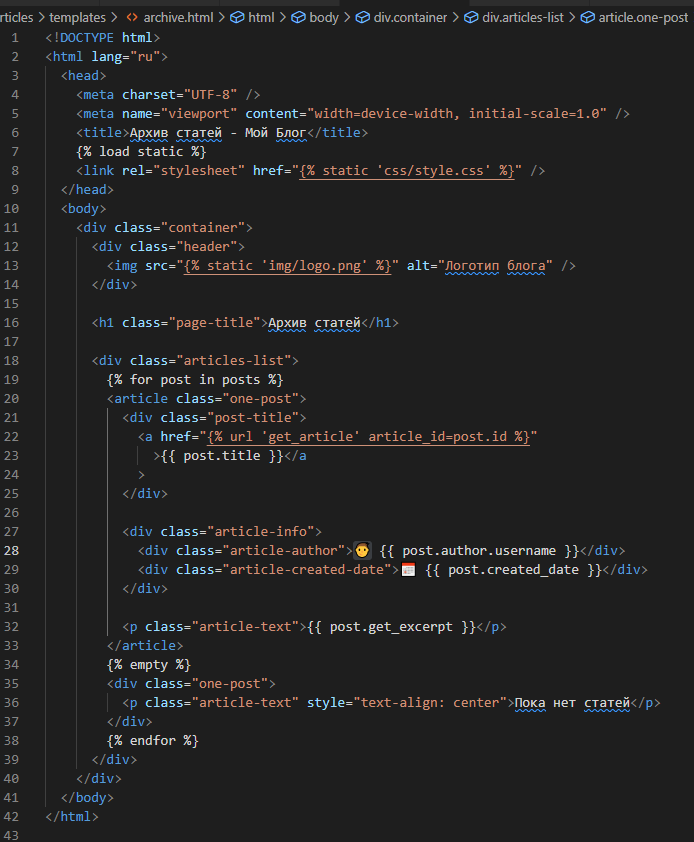


Рисунок 8. Создание и редактирование файла archive.html

Рисунок 9. Создание шаблона для отдельной статьи

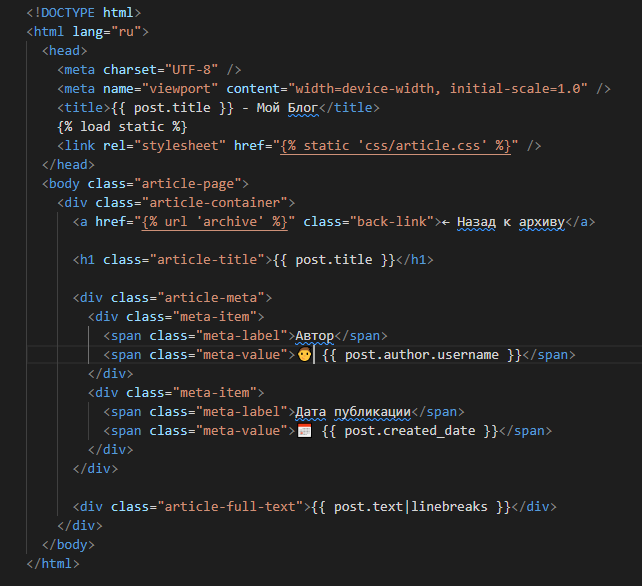


Рисунок 9. Создание шаблона для отдельной статьи

Файл articles/views.py нужен для обработки логики представления в Django. Это контроллер, который связывает модели базв данных с шаблонами Html.

Рисунок 10 – редактирование views.py

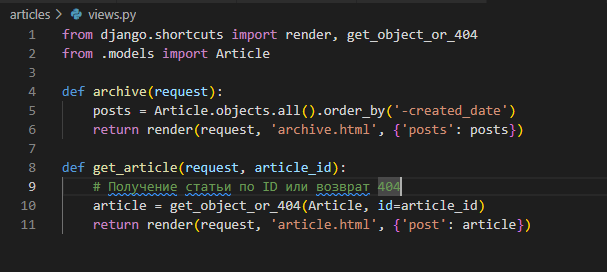


Рисунок 10 – редактирование views.py

Настройка URL-адреса внутри приложения articles. В папке articles был создан файл urls.py

Рисунок 11 редактирование файла urls.py

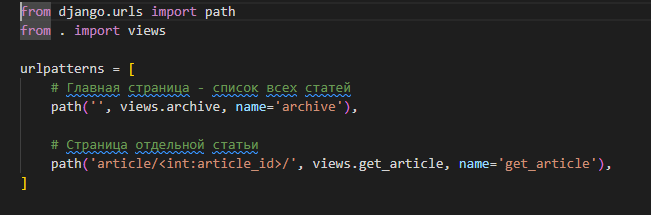


Рисунок 11 редактирование файла urls.py

Редактирование файла urls.py, находящегося в каталоге blog. Передача главному файлу urls.py проекта информации о том, что все запросы к корню сайта (/) должны перенаправляться в приложение articles.

Рисунок 12 - Редактирование файла urls.py

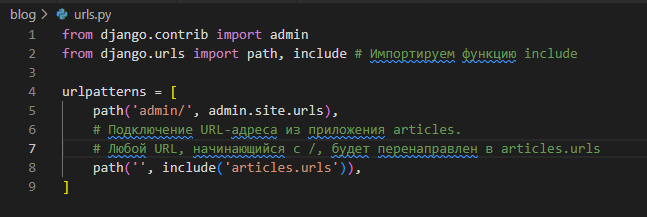


Рисунок 12 - Редактирование файла urls.py

Оформление блока. В каталоге blog, в папке static/css были добавлены css файлы оформления. Файл style.css для общих стилей, файл article.css для оформления отдельной статьи. Рисунок 13 – добавление css файлов оформления блога.

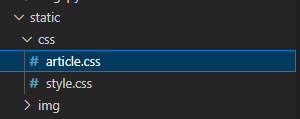


Рисунок 13 – добавление css файлов оформления блога.

Рисунок 14. Редактирование файла style.css

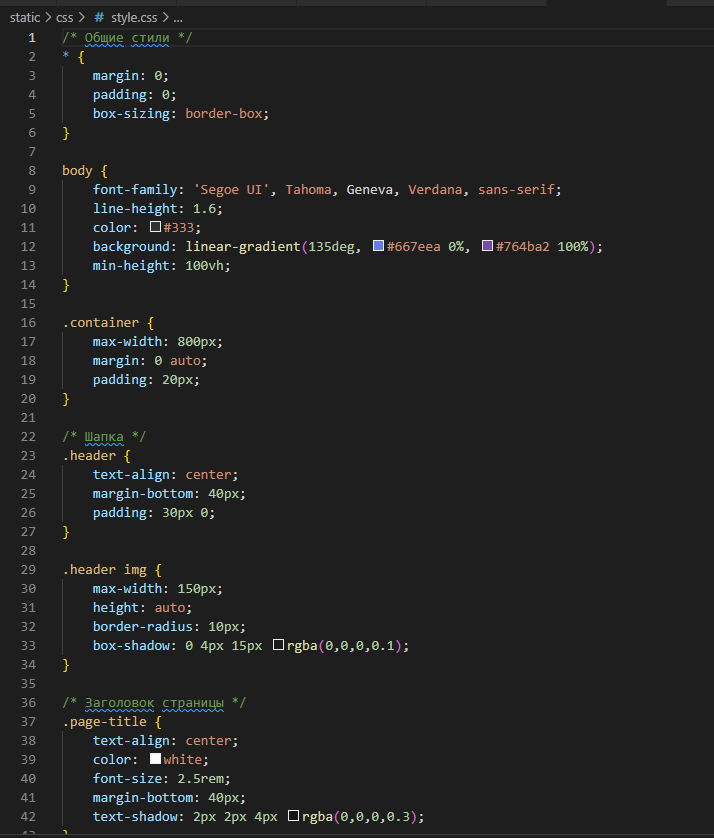


Рисунок 14. Редактирование файла style.css

Рисунок 15. Редактирование файла article.css

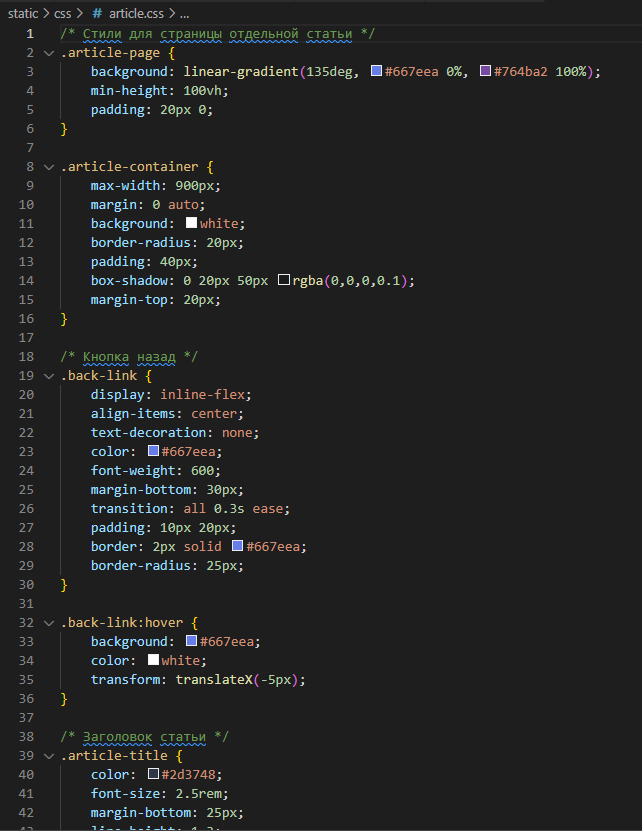


Рисунок 15. Редактирование файла article.css

Обновим страницу блока. Рисунок 16

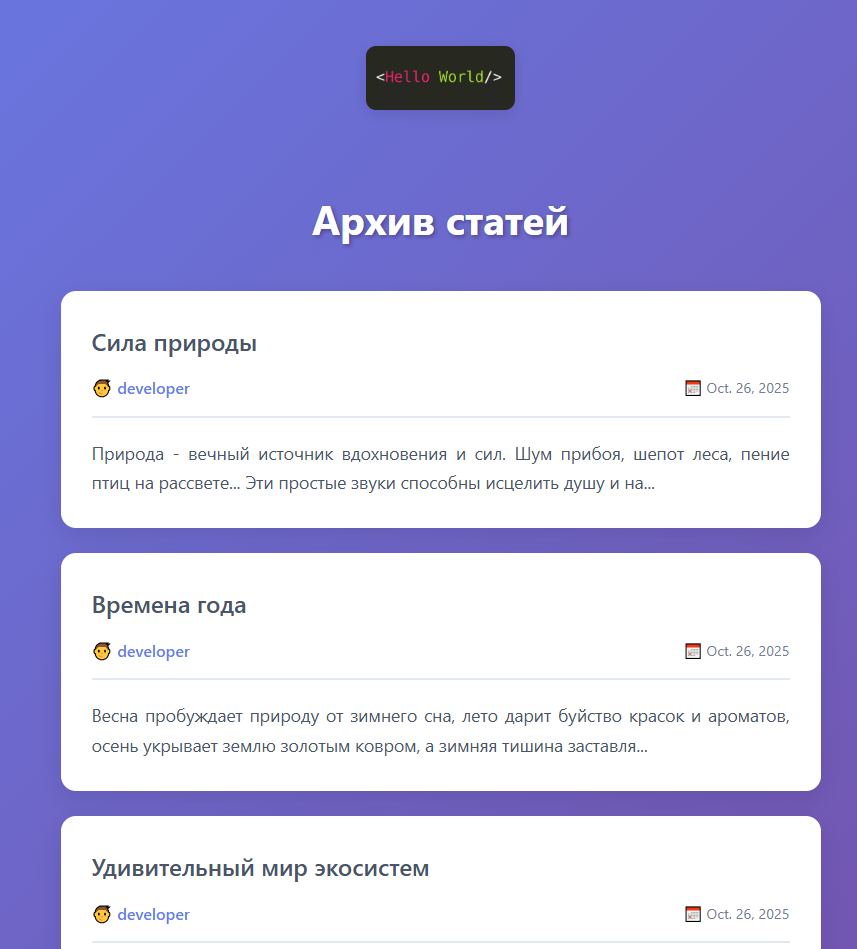


Рисунок 16 – основная страница блога

Рисунок 17 – оформление отдельной статьи



Рисунок 17 – оформление отдельной статьи