

Университет ИТМО

Факультет «Инфокоммуникационных технологий»
Направление подготовки «название направления подготовки»

Лабораторная работа №8
«Разработка проекта на django»

Выполнил:
Кузнецов Никита Сергеевич
Группа К33212
Проверил:
Марченко Елена Вадимовна

Санкт-Петербург
2023

Цель работы:

Ход работы:

1 задание: рассмотреть (установить, посмотреть в работе) готовый проект на django.

Для выполнения данного задания было создано виртуальное окружение, был скопирован проект из архива к лекции и запущен следующими двумя командами:

- 1) python example_project/manage.py migrate
- 2) python example_project/manage.py runserver

В результате чего можно было открыть готовый сайт по адресу 127.0.0.1:8000 (рис. 1).

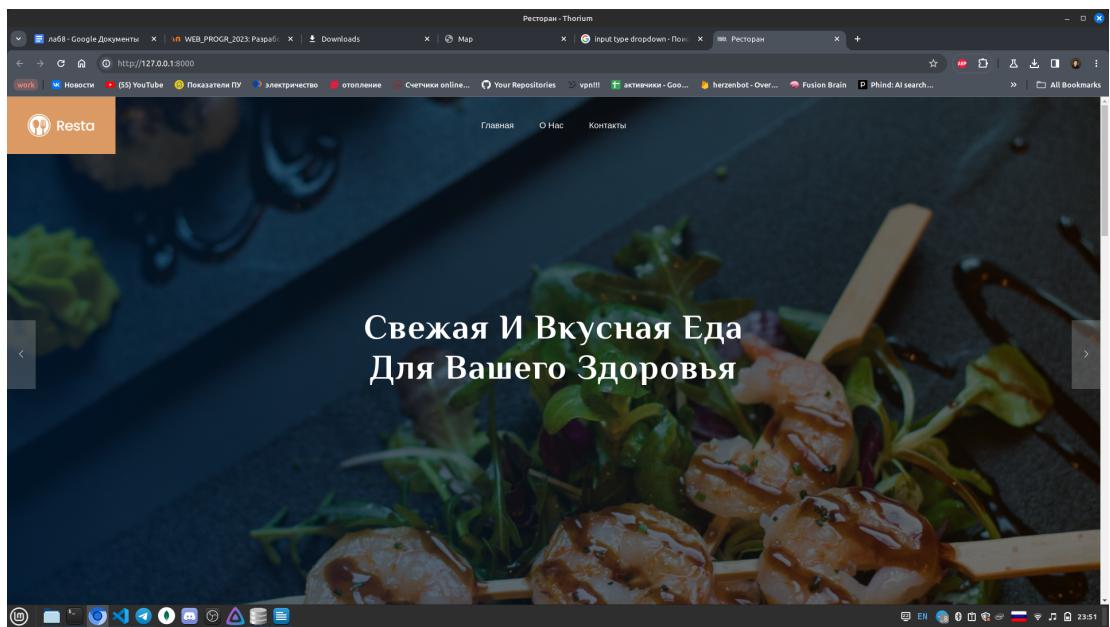


Рисунок 1 - Сайт-пример из архива к лекции.

2 задание: создать свой проект на django с нуля. В проекте три страницы: главная, о разработке, страница обратной связи. У всех трех страниц одинаковые футеры и хидеры (сделать шаблоны в django), с использованием возможностей django разработать форму обратной связи (на третьей странице) и обработку данных формы.

Для выполнения данного задания был создан проект Django с названием myblog. Внутри проекта было создано приложение myblogsite, внутри которого будет находиться логика и интерфейс будущего сайта.

Чтобы запустить приложение сначала необходимо выполнить два предварительных действия:

- 1) Провести миграции базы данных для пользователей и групп админ-панели (рис. 2)
- 2) Создать суперпользователя через командную строку и команду createsuperuser (рис. 3)

```
● (venv) crawlic@crawlic-pc:~/Desktop/code/web-programming-itmo/lab8/myblog$ 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 2 - Запуск миграций для стандартных моделей админ-панели Django.

```
● (venv) crawlic@crawlic-pc:~/Desktop/code/web-programming-itmo/lab8/myblog$ python manage.py createsuperuser
Username (leave blank to use 'crawlic'): admin
Email address: admin@mail.ru
Password:
Password (again):
The password is too similar to the username.
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.
○ (venv) crawlic@crawlic-pc:~/Desktop/code/web-programming-itmo/lab8/myblog$ █
```

Рисунок 3 - Создание суперпользователя для админ-панели.

Далее, можно запустить проект с помощью команды runserver (рис. 4).

```
(venv) crawlic@crawlic-pc:~/Desktop/code/web-programming-itmo/lab8/myblog$ python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
December 03, 2023 - 14:07:08
Django version 4.2.7, using settings 'myblog.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CONTROL-C.
```

Рисунок 4 - Запуск сервера с пустым проектом Django.

После всех выполненных команд по корневой ссылке будет открываться приветственный экран для проекта Django (рис. 5), а по ссылке /admin/ будет экран входа (рис. 6) для ранее созданного суперпользователя.

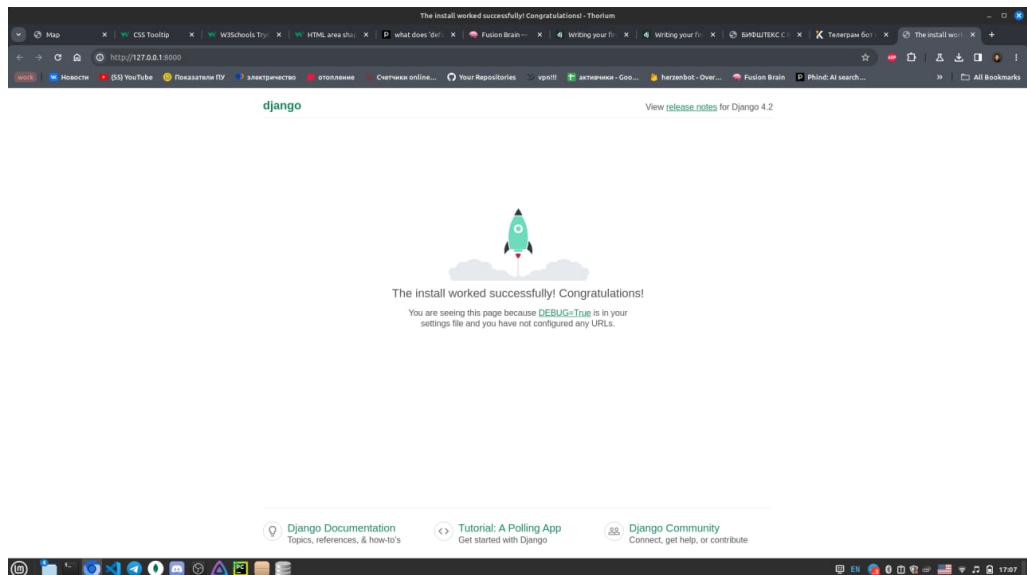


Рисунок 5 - Приветственная страница пустого проекта на Django.

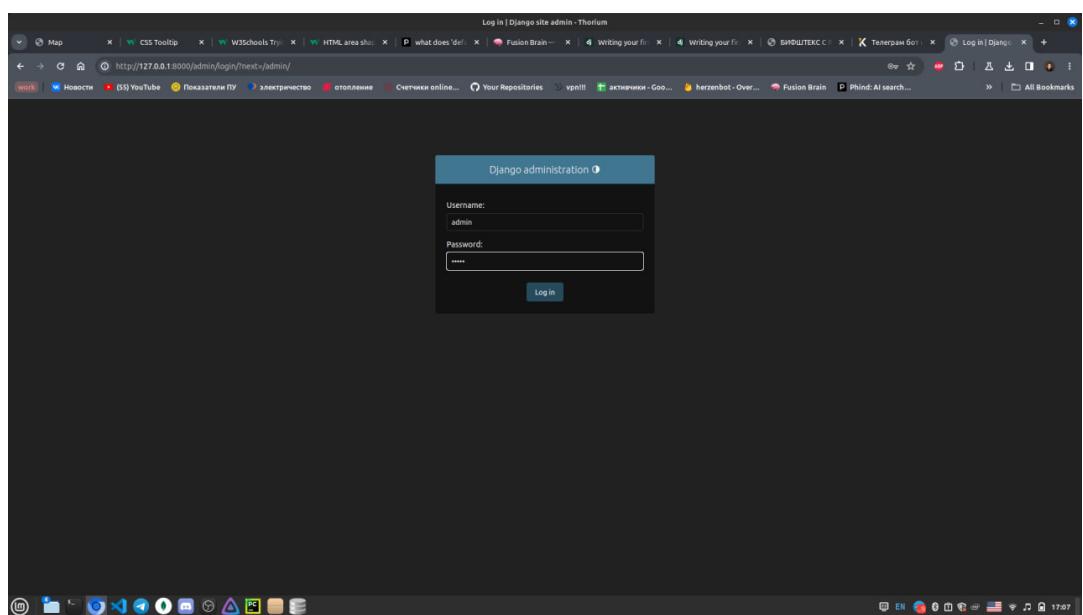


Рисунок 6 - Вход в админ-панель проекта на Django.

В админ-панели пустого проекта существуют две сущности - Users и Groups (рис. 7), которые отвечают за ранее упомянутых суперпользователей и их группы с различными правами доступа.

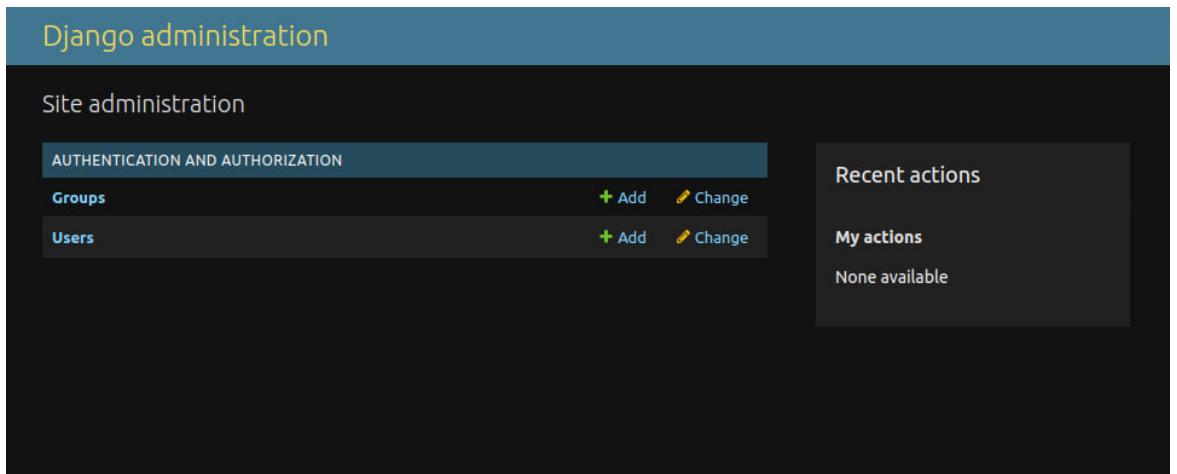


Рисунок 7 - Админ-панель пустого проекта на Django.

Перед тем как начать разработку сайта остался последний шаг - добавление приложения в список INSTALLED_APPS в файле settings.py в папке myblog (рис. 8). Это действие позволяет Django понять, что приложение добавлено в общий проект и его нужно отслеживать.

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

Рисунок 8 - Добавление приложения myblogsite в список установленных в проекте приложений

В основе идеи для разработки сайта лежало простое приложение для просмотра статей от авторов блога. Для реализации данного функционала была создана модель Article (приложение 1) со следующими полями:

- title - заголовок статьи
- text - содержимое статьи
- image - изображение для статьи

- dt - дата и время публикации
- category - категория статьи

Для просмотра статей были созданы два отображения в файле views.py (приложение 2):

- 1) index - просмотр всех статей из базы данных
- 2) article_view - просмотр конкретной статьи

Для отправки обратной связи от пользователя была создана форма Django в файле forms.py (приложение 3) со следующими полями:

- name - имя отправителя
- email - почта отправителя
- message - сообщение от пользователя

Для показа главной страницы, страницы с информацией и обратной связью были созданы соответствующие файлы HTML с элементами от шаблонизатора jinja2, чтобы иметь общую структуру и вид у всех страниц:

- main.html - шаблон для всех страниц сайта, в котором находятся футер и хедер (приложение 4)
- index.html - файл с содержимым корневой страницы, на которой отображаются все статьи (приложение 5)
- article.html - файл с отображением полной статьи (приложение 6)
- contacts.html - файл с формой обратной связи (приложение 7)
- about.html - файл с информацией о сайте (приложение 8)

После написания основной логики приложения и всех отображений были запущены миграции (рис. 9), а также добавлено отображение Article для админ-панели (рис. 10-11), чтобы можно было создавать статьи для отображения на сайте.

```
● (venv) crawlic@crawlic-pc:~/Desktop/code/web-programming-itmo/lab8/myblog$ python manage.py makemigrations
  Migrations for 'myblogsite':
    myblogsite/migrations/0001_initial.py
      - Create model Article
○ (venv) crawlic@crawlic-pc:~/Desktop/code/web-programming-itmo/lab8/myblog$ █
```

Рисунок 9 - Создание миграций для модели Article.

```
from django.contrib import admin
from models import Article
admin.site.register(Article)
```

Рисунок 10 - Добавление модели Article в админ-панель внутри кода.

The screenshot shows the Django administration interface. At the top, it says "Django administration" and "Site administration". Below that, there's a sidebar with "Recent actions" (empty) and "My actions" (empty). The main area has a dark header with "AUTHENTICATION AND AUTHORIZATION" containing "Groups" and "Users" with "Add" and "Change" buttons. Below that is a header for "MYBLOGSITE" containing "Articles" with "Add" and "Change" buttons. The "Articles" section is currently empty.

Рисунок 11 - Отображение модели Article внутри интерфейса админ-панели.

После этих действий, можно написать первую статью (рис. 12), увидеть ее на главной странице сайта (рис. 13) и прочитать на странице с полным текстом статьи (рис. 14).

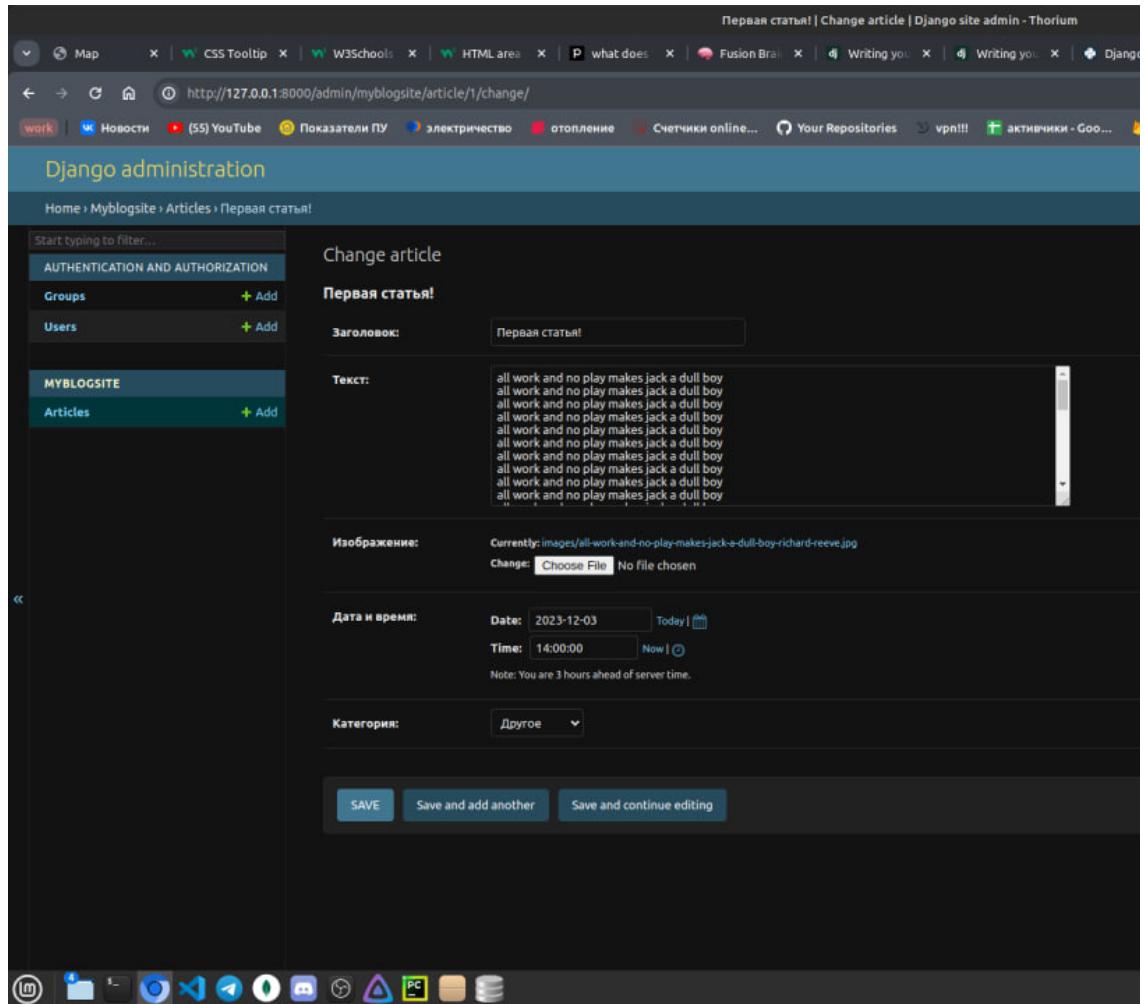


Рисунок 12 - Создание новой статьи в админ-панели.

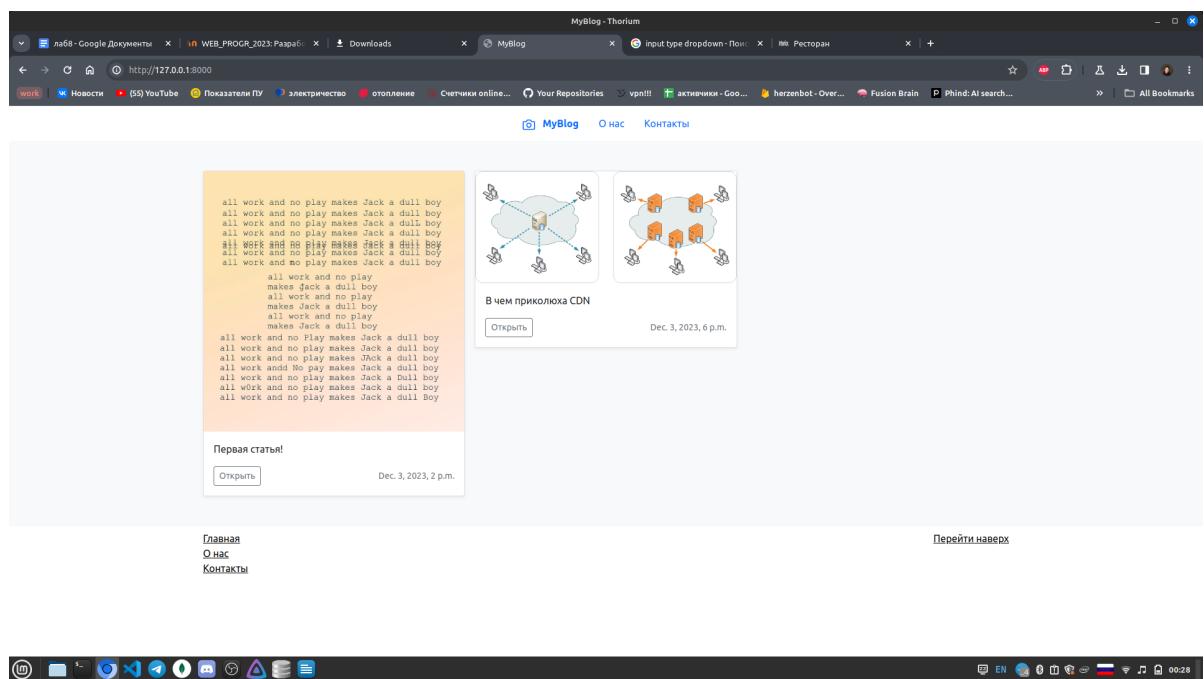
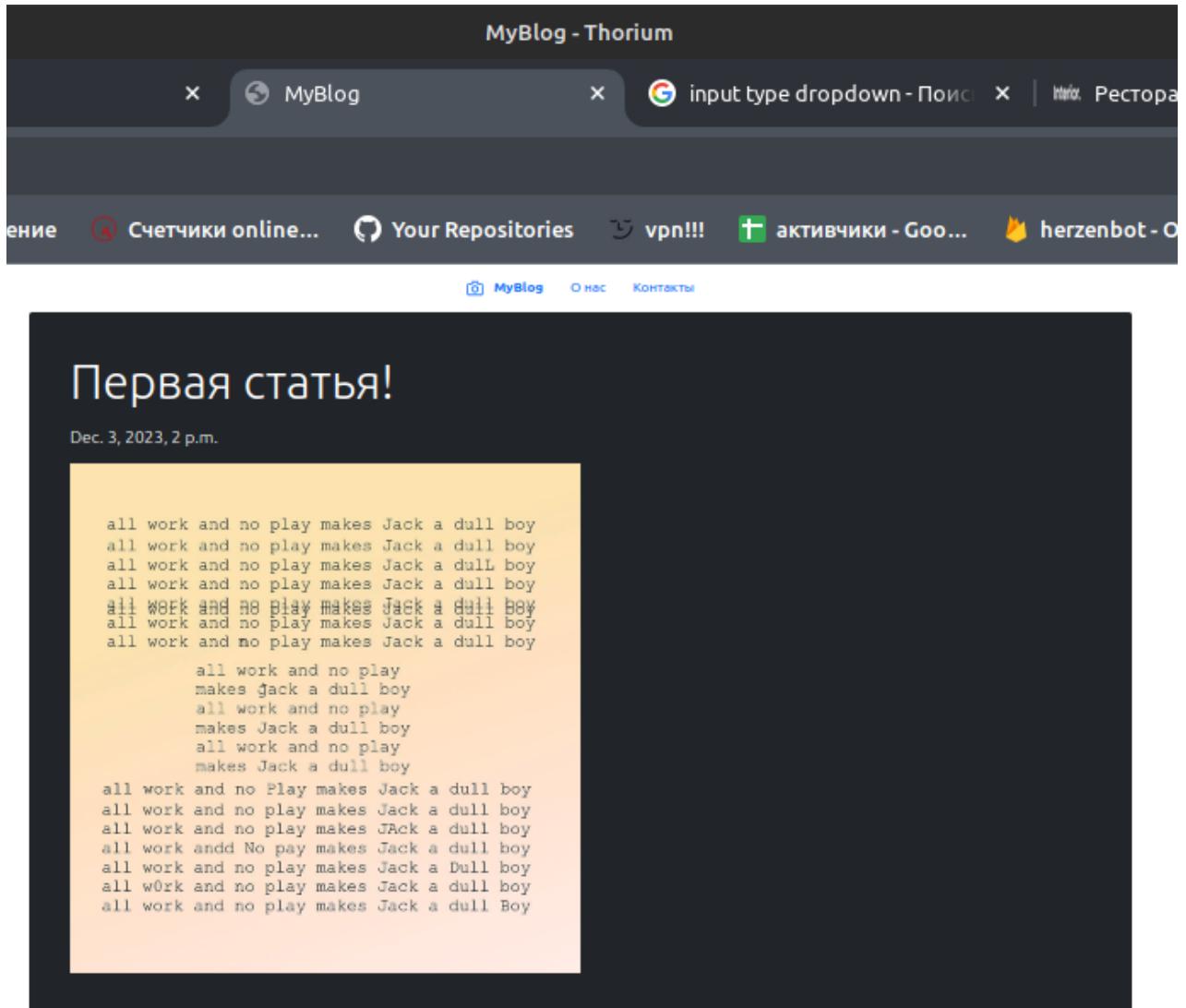


Рисунок 13 - Отображение созданной статьи на главном экране.



Первая статья!

[Главная](#)
[О нас](#)
Контакты

[Перейти наверх](#)

Рисунок 14 - Просмотр статьи на сайте.

Также, можно просмотреть информацию о сайте в разделе “О нас” (рис. 15) и отправить обратную связь в разделе “Контакты” (рис. 16).

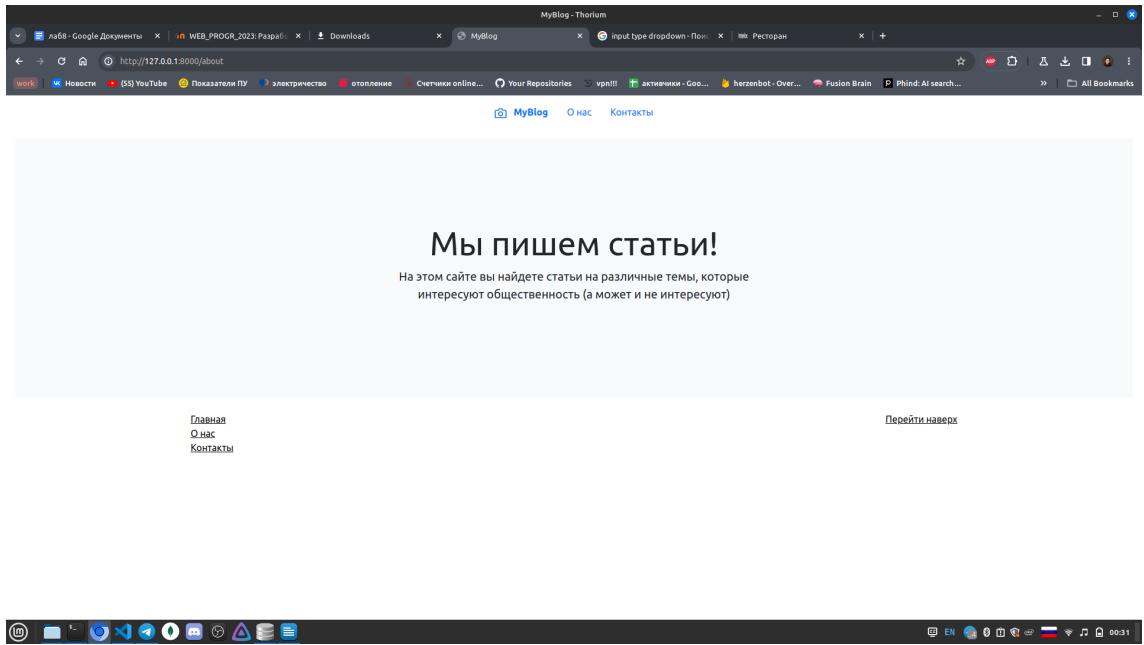


Рисунок 15 - Раздел сайта “О нас”.

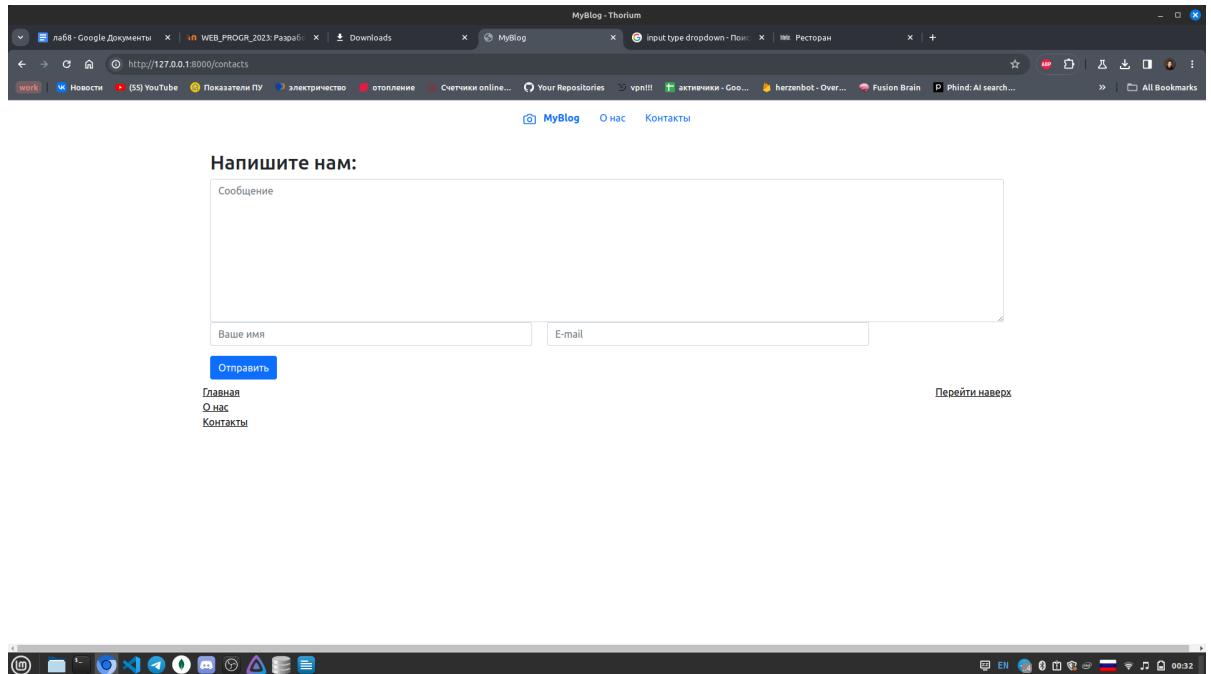


Рисунок 16 - Раздел сайта “Контакты”.

Вывод:

В результате работы был создан простой проект на Django с нуля. Были использованы встроенные технологии фреймворка, такие как: модели, формы, представления и шаблоны для отображения, создания и редактирования данных, а также была написана логика их обработки и поведения на языке Python.

Приложение 1. Файл models.py с моделью Article.

```
from django.db import models

class Article(models.Model):
    title = models.CharField(max_length=256, verbose_name="Заголовок")
    text = models.TextField(verbose_name="Текст")
    image = models.ImageField(upload_to="images", verbose_name="Изображение")
    dt = models.DateTimeField(verbose_name="Дата и время")

    CATEGORIES = [
        ("SPORTS", "Спорт"),
        ("IT", "IT"),
        ("ANIMALS", "Животные"),
        ("OTHER", "Другое"),
    ]

    category = models.CharField(
        choices=CATEGORIES, default="IT", verbose_name="Категория", max_length=64
    )

    def __str__(self):
        return self.title, self.image
```

Приложение 2. Файл views.py с представлениями данных для страниц сайта.

```
from django.shortcuts import render
from django.forms.models import model_to_dict

from .forms import ContactForm
from .models import Article


def index(request):
    context = {"articles": Article.objects.all()}
    return render(request, "index.html", context=context)

def about(request):
    return render(request, "about.html")

def contacts(request):
    context = {}
    if request.method == "POST":
        form = ContactForm(request.POST)
        if form.is_valid():
            context = {"success": 1}
    else:
        form = ContactForm()
    context["form"] = form
    return render(request, "contacts.html", context=context)

def article_view(request, pk: int):
    article = Article.objects.get(pk=pk)
    return render(request, "article.html", model_to_dict(article))
```

Приложение 3. Файл forms.py с представлением и валидацией для формы обратной связи.

```
from django import forms

class ContactForm(forms.Form):
    name = forms.CharField(
        min_length=2,
        widget=forms.TextInput(
            attrs={"placeholder": "Ваше имя", "class": "form-control valid"})
    ),
    email = forms.EmailField(
        widget=forms.EmailInput(
            attrs={"placeholder": "E-mail", "class": "form-control valid"})
    )
    message = forms.CharField(
        min_length=20,
        widget=forms.Textarea(
            attrs={
                "placeholder": "Сообщение",
                "cols": 30,
                "rows": 9,
                "class": "form-control w-100",
            }
        ),
    )
```

Приложение 4. Файл main.html для шаблонизации остальных страниц сайта.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>MyBlog</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztQTwFspd3yD65VohhpooCOnLASjC" crossorigin="anonymous">
    <style>
        a {color: black;}
    </style>
</head>
<body>
    <header class="mb-auto">
        <div>
            <nav class="navbar nav-masthead justify-content-center">
                <a class="nav-link active" aria-current="page" href="/">
                    <img alt="MyBlog logo" data-bbox="158 418 868 524" />
                    <strong>MyBlog</strong>
                </a>
                <a class="nav-link" href="/about">О нас</a>
                <a class="nav-link" href="/contacts">Контакты</a>
            </nav>
        </div>
    </header>
    <main>
        {%
            block content
        %}
    </main>
    <footer class="text-muted py-2">
        <div class="container">
            <p class="float-end mb-1">
                <a href="#">Перейти наверх</a>
            </p>
            <p class="mb-0"><a href="/">Главная</a></p>
            <p class="mb-0"><a href="/about">О нас</a></p>
            <p class="mb-0"><a href="/contacts">Контакты</a></p>
        </div>
    </footer>
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js" integrity="sha384-MrcW6ZMFYlzcLA8Nl+TtU9y/fsvAd8+nEremaTC1HtCgQiKx9vF" crossorigin="anonymous"></script>
</body>
```

```
</html>
```

Приложение 5. Файл index.html для просмотра основной страницы сайта.

```
{% extends 'main.html' %}  
{% load static %}  
{% block content %}  
    <div class="album py-5 bg-light">  
        <div class="container">  
            <div class="row row-cols-1 row-cols-sm-2 row-cols-md-3 g-3">  
                {% for a in articles %}  
                    <div class="col">  
                        <div class="card shadow-sm">  
                              
                            <div class="card-body">  
                                <p class="card-text">{{ a.title }}</p>  
                                <div class="d-flex justify-content-between align-items-center">  
                                    <div class="btn-group">  
                                        <button type="button" class="btn btn-sm btn-outline-secondary"  
onclick="location.href = '/articles/{{ a.id }}';">Открыть</button>  
                                    </div>  
                                    <small class="text-muted">{{ a.dt }}</small>  
                                </div>  
                            </div>  
                        </div>  
                    </div>  
                {% endfor %}  
            </div>  
        </div>  
    </div>  
{% endblock %}
```

Приложение 6. Файл article.html для просмотра подробной информации о статье.

```
{% extends 'main.html' %}  
{% load static %}  
{% block content %}  
<div class="container">  
    <div class="p-4 p-md-5 mb-4 text-white rounded bg-dark">  
        <div class="col-md-6 px-0">  
            <h1 class="display-4">{{ title }}</h1>  
            <p class="lead my-3">{{ dt }}</p>  
              
        </div>  
</div>  
  
<div class="row g-5">  
    <div class="col-md-8">  
        <h3 class="pb-4 mb-4 border-bottom">  
            {{title}}  
        </h3>  
  
        <article class="blog-post">  
            <p>{{ text }}</p>  
        </article>  
    </div>  
</div>  
</div>  
{% endblock %}
```

Приложение 7. Файл contacts.html с формой обратной связи.

```
{% extends 'main.html' %}  
{% load static %}  
{% block content %}  
<br>  
<div class="row">  
    <div class="container col-md-8">  
        <h2>Напишите нам:</h2>  
    </div>  
    <div class="container col-md-8">  
        <form class="form-contact contact_form" action="" method="post"  
novalidate="novalidate">  
            {% csrf_token %}  
            {% if success %}  
                <div class="alert alert-success">  
                    <strong>Спасибо за сообщение!</strong>  
                </div>  
            {% else %}  
                <div class="row">  
                    <div class="col-12">  
                        <div class="form-group">  
                            {{ form.message }}  
                        </div>  
                        {% if form.message.errors %}  
                            <div class="alert alert-danger">  
                                <strong>{{ form.message.errors }}</strong>  
                            </div>  
                        {% endif %}  
                    </div>  
                    <div class="col-sm-5">  
                        <div class="form-group">  
                            {{ form.name }}  
                        </div>  
                        {% if form.name.errors %}  
                            <div class="alert alert-danger">  
                                <strong>{{ form.name.errors }}</strong>  
                            </div>  
                        {% endif %}  
                    </div>  
                    <div class="col-sm-5">  
                        <div class="form-group">  
                            {{ form.email }}  
                        </div>  
                        {% if form.email.errors %}  
                            <div class="alert alert-danger">  
                                <strong>{{ form.email.errors }}</strong>  
                            </div>  
                        {% endif %}  
                    </div>  
                </div>  
                <div class="form-group mt-3">  
                    <button type="submit" class="btn btn-primary">Отправить</button>  
                </div>  
            {% endif %}  
        </form>  
    </div>  
</div>
```

```
        </div>
    {%
    endif %
}
</form>
</div>
{%
endblock %
}
```

Приложение 8. Файл about.html для просмотра информации о сайте.

```
{% extends 'main.html' %}  
{% load static %}  
{% block content %}  
  
<div class="position-relative overflow-hidden p-3 p-md-5 m-md-3 text-center bg-light">  
    <div class="col-md-5 p-lg-5 mx-auto my-5">  
        <h1 class="display-4 fw-normal">Мы пишем статьи!</h1>  
        <p class="lead fw-normal">На этом сайте вы найдете статьи на различные темы, которые  
интересуют общественность (а может и не интересуют)</p>  
    </div>  
    <div class="product-device shadow-sm d-none d-md-block"></div>  
    <div class="product-device product-device-2 shadow-sm d-none d-md-block"></div>  
</div>  
  
{% endblock %}
```