

Проектно-исследовательская работа на тему «Сайт для экотуризма в России»

Обучающиеся 10Б ГБОУ Школы №1748:

Смирнов Александр Денисович

Андрияшкин Владимир Сергеевич

Мамонтов Ростислав Валерьевич

Руководитель от Университета:

Гришина Арина Александровна

Москва 2025 г.

СОДЕРЖАНИЕ

1) ВВЕДЕНИЕ

1. Актуальность работы
2. Цель и задачи

2) ОСНОВНАЯ ЧАСТЬ

1. Разработка структуры и дизайна
2. Создание программного обеспечения и верстки
3. Основные разделы сайта и база данных
4. Тестирование и комментарии

3) ЗАКЛЮЧЕНИЕ

4) СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

ВВЕДЕНИЕ: Актуальность работы

В современном мире все больше людей стремятся проводить время на природе, наслаждаться её красотой и гармонией. Проект направлен на создание удобной онлайн-платформы, которая поможет пользователям находить информацию о природных достопримечательностях нашей страны, получать полезные советы по организации экологических туров.

Статистика развития экотуризма по данным Росзаповедцентра, Ростуризма, Минприроды РФ изображена на рисунке 1.

Поток экотуристов (млн. чел) в России

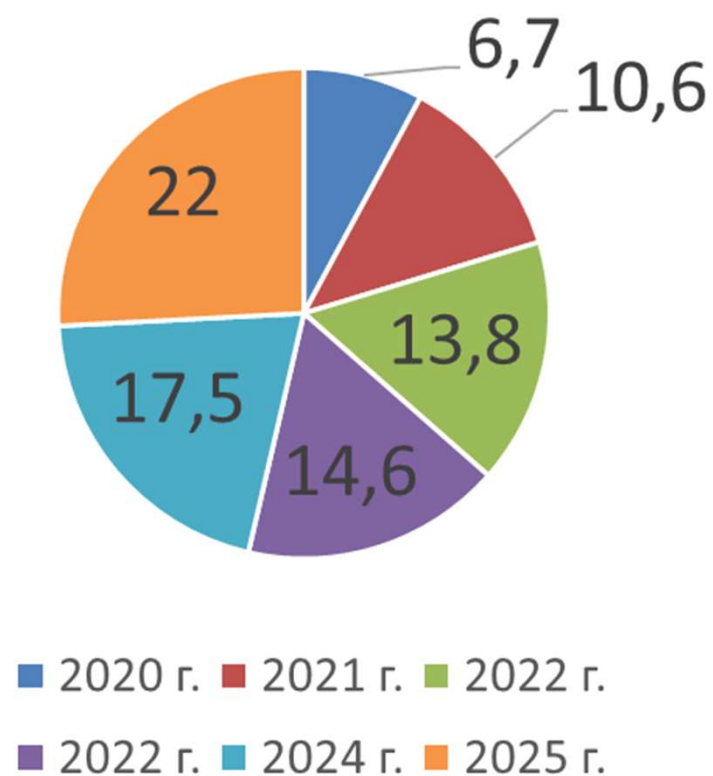


Рисунок 1 — Поток экотуристов (млн. чел) в России

ВВЕДЕНИЕ: Цель и задачи

Цель:

Создать сайт, который будет предоставлять информацию об уникальных местах природы России и знакомить с новостями в сфере экотуризма.

Задачи работы:

- 1) Разработать структуру и дизайн сайта.
- 2) Создать верстку веб-страниц, то есть подготовить Front-end.
- 3) Разработать Back-end, то есть провести работу с базами данных.
- 4) Провести функциональное и оценочное тестирование.

ОСНОВНАЯ ЧАСТЬ: Разработка структуры

Составляющие:

- Популярные точки экотуризма (краткое и подробное описание)
- Новости
- Обновления
- Регистрация
- Поддержка

Виды диаграмм:

- ER-диаграмма (схема «сущность-связь»)
- Диаграмма состояний (описание поведения систем)

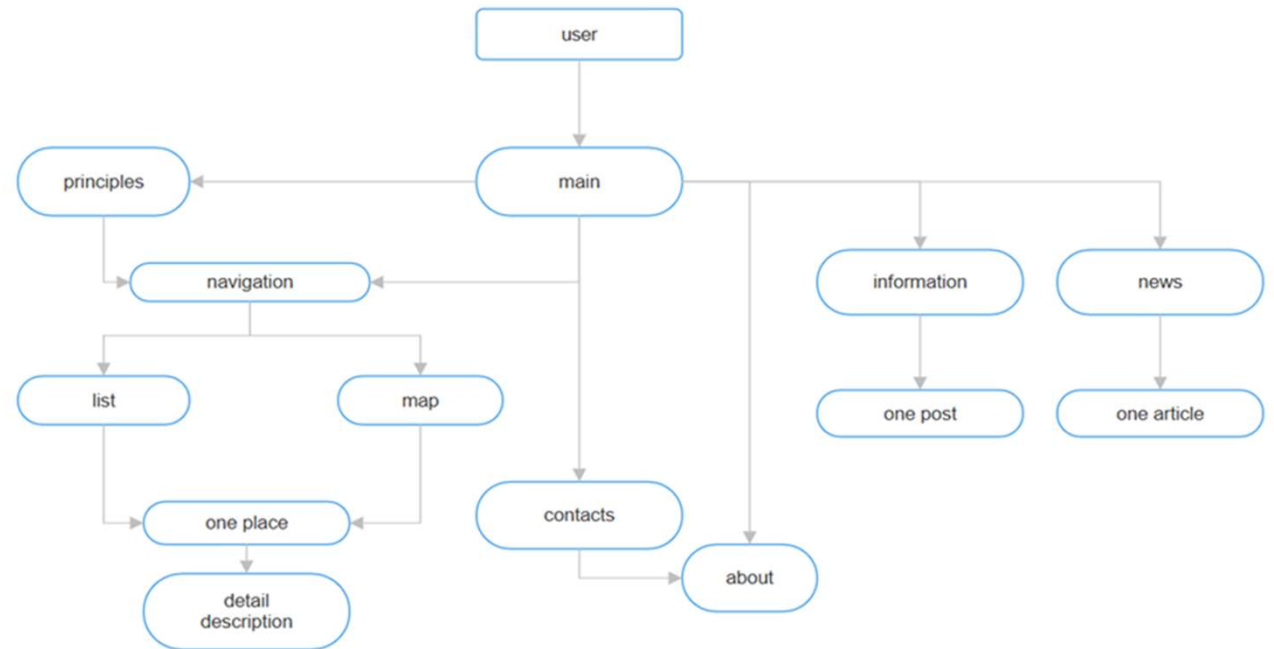
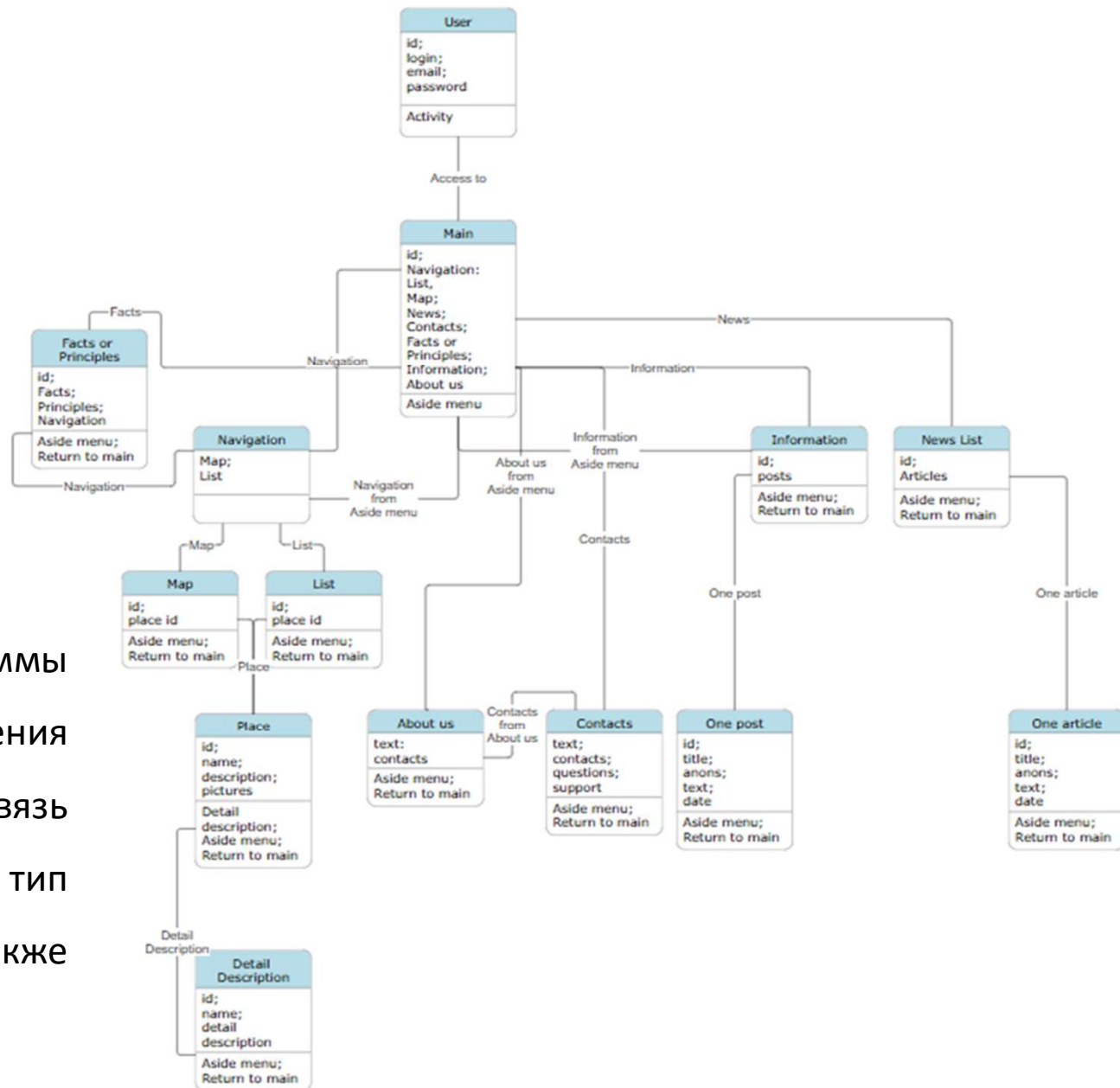


Рисунок 2 — Концептуальная диаграмма компоновки страницами

ОСНОВНАЯ ЧАСТЬ: Разработка структуры

Рисунок 3 — Подробная
схема с учетом типа
содержания страницы или
информации вывода из базы
данных.

Упрощенная версия диаграммы
показывает все возможные перемещения
пользователя по страницам и их связь
между собой, а подробная учитывает тип
информации или содержание, а также
информацию вывода из базы данных.



ОСНОВНАЯ ЧАСТЬ:

Разработка дизайна

Выбор графического редактора Figma.

Этот графический редактор был выбран в силу следующих основных характеристик:

- Онлайн-доступ
- Совместная работа
- Интеграции
- Библиотеки компонентов

Создание интерфейса по требованиям UX и UI:

UX —

- Точное расположение объектов на странице, то есть по единому шаблону каждая страница имеет порядок или структуру.

UI —

- Страницы сайта должны сочетать в себе основные цвета зеленых оттенков, гармонирующих с природой.
- Картинки изображают пейзаж местности, о которой идет речь. Это можно увидеть на примере главной страницы и карты.

ОСНОВНАЯ ЧАСТЬ: Разработка дизайна

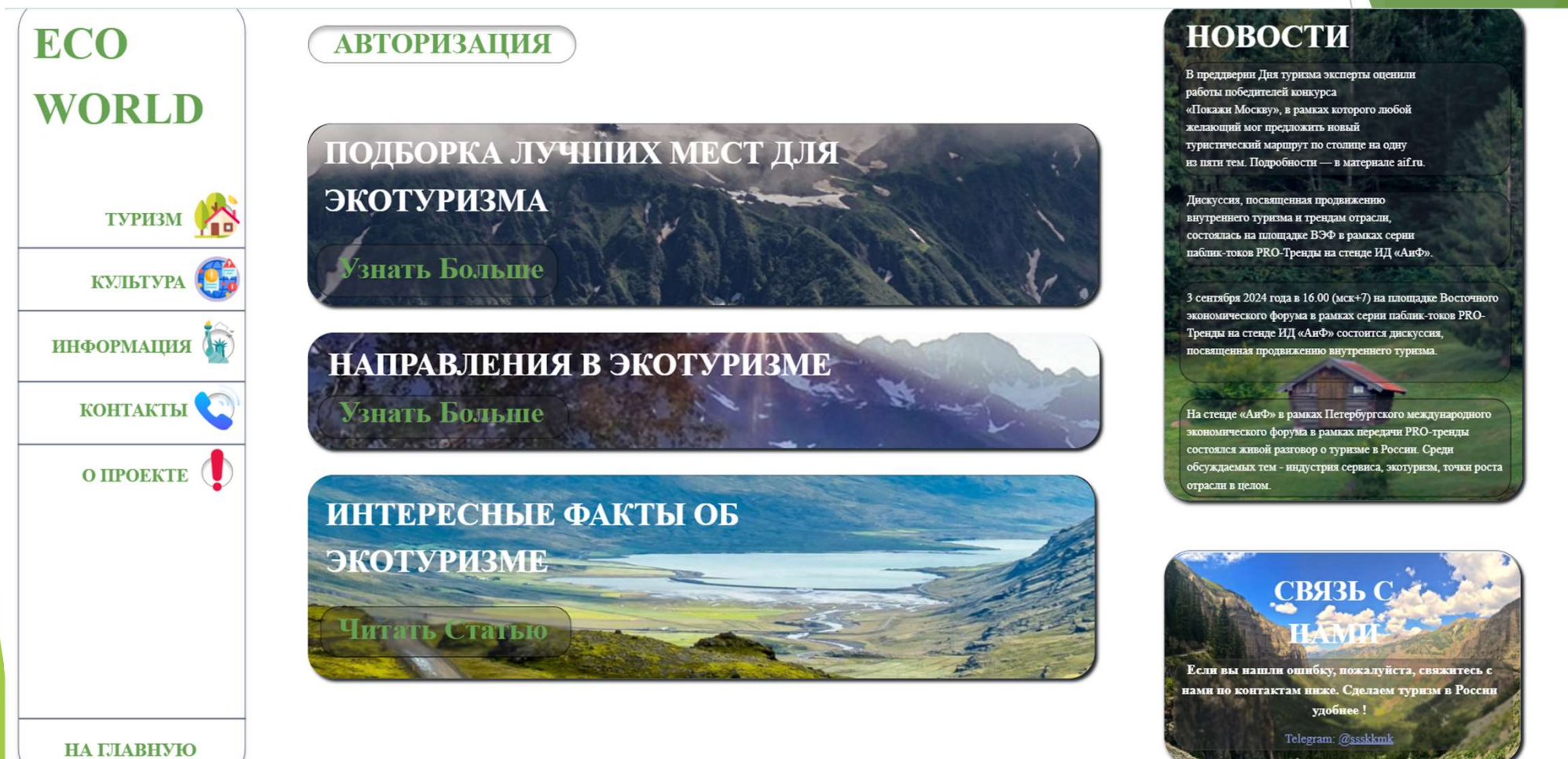


Рисунок 4 — Главная страница

ОСНОВНАЯ ЧАСТЬ: Разработка дизайна

ECO WORLD

ТУРИЗМ 

КУЛЬТУРА 

ИНФОРМАЦИЯ 

КОНТАКТЫ 

ПРОЕКТ 

[НА ГЛАВНУЮ](#)

Горный Алтай

О Месте

Горный Алтай — это один из самых красивых и уникальных регионов России, который привлекает туристов своими живописными ландшафтами, богатым биоразнообразием и культурным наследием. Этот регион идеально подходит для экотуризма благодаря своей нетронутой природе и множеству возможностей для активного отдыха.

[Узнать про экотуризм](#)



Рисунок 5 — Пример страницы о месте или достопримечательности

ОСНОВНАЯ ЧАСТЬ: Разработка дизайна

КАРТА

Рисунок 6 — Карта

ECO
WORLD

ТУРИЗМ 

КУЛЬТУРА 

ИНФОРМАЦИЯ 

КОНТАКТЫ 

ПРОЕКТ 

[НА ГЛАВНУЮ](#)



ОСНОВНАЯ ЧАСТЬ: Создание программного обеспечения и верстки

Среда разработки Visual Studio Code с использованием фреймворка Django:

- Разнообразие языков программирования
- Управление версиями
- Расширяемость
- Интеллектуальная помощь (GitHub Copilot, Blackbox, ChatGPT, Tabnine, Mintlify Doc Writer, Kodezi)

Фреймворк Django:

Django — свободный фреймворк для веб-приложений на языке Python. С его помощью можно быстрее и проще реализовывать на Python сайты и приложения.

Верстка сайта и принципы строения каждой страницы:

Верстка — это соединение и расположение на странице документа разных элементов веб-сайта: текстовых блоков, изображений, таблиц, видео и т.д.

ОСНОВНАЯ ЧАСТЬ: Верстка

Рисунок 7 — Шаблон страницы

my_site > main > templates > main > layout.html

```
1  {% load static%}
2  <!DOCTYPE html>
3  <html lang = "ru">
4  <head>
5      <meta charset="utf-8">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <meta http-equiv="X-UA-Compatible" content = "IE-edge">
8      <title>{%block title%} {%endblock %}</title>
9      <link rel = 'stylesheet' href = "https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css">
10     <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/@fortawesome/fontawesome-free@6.6.0/css/fontawesome.min.css"
11     integrity="sha384-NvKbDTEnL+A8F/AA5Tc5kmMLSJHU0868P+ldTtpJJieQdGYaUIuLr4lVGOEA10cMy" crossorigin="anonymous">
12     <link rel="apple-touch-icon" sizes="180x180" href="static/main/img/apple-touch-icon.png">
13     <link rel="icon" type="image/png" sizes="32x32" href="static/main/img/favicon-32x32.png">
14     <link rel="icon" type="image/png" sizes="16x16" href="static/main/img/favicon-16x16.png">
15     <link rel="manifest" href="static/main/img/site.webmanifest">
16     {%block css%}{%endblock %}
17     <style>
18         {%block style%}
19         {%endblock %}
20     </style>
21 </head>
22 <body>
23     {%block body%} {%endblock %}
24     <script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.11.8/dist/umd/popper.min.js"
25     integrity="sha384-I7E8VVD/ismYTF4hNIPjVp/Zjvgyol6VFvRkX/vR+Vc4jQkC+hVqc2pM80Dewa9r" crossorigin="anonymous"></script>
26     <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"
27     integrity="sha384-YvpcrYf0tY3lHB60NNkmXc5s9fDVZLESaAA55NDzOxhy9GkcIdslK1eN7N6jIeH2" crossorigin="anonymous"></script>
28 </body>
29 </html>
```

ОСНОВНАЯ ЧАСТЬ: Базы данных

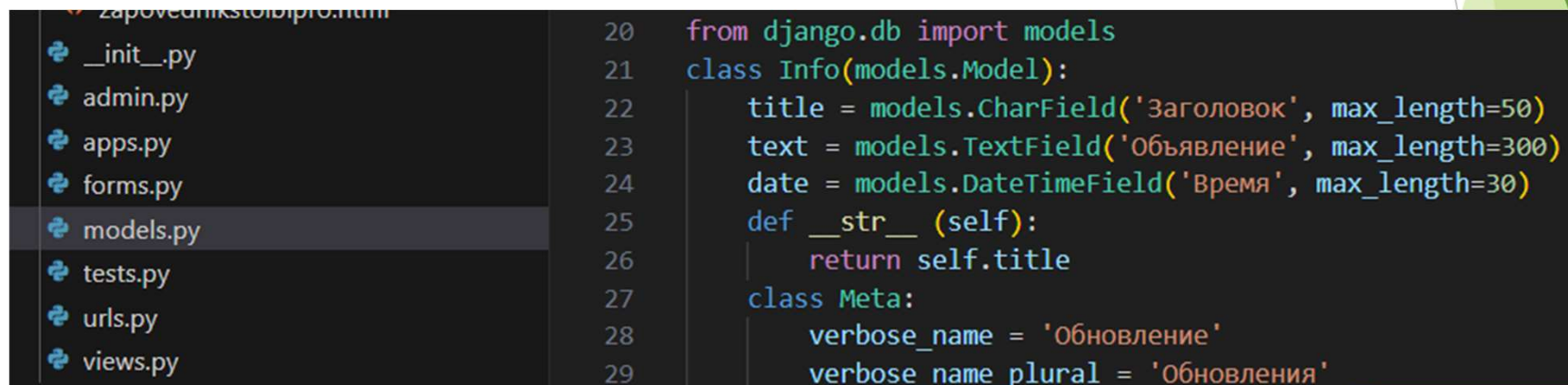
База данных — набор информации, которая хранится упорядоченно в электронном виде; набор данных, который как-то структурирован. Базу данных нельзя назвать программой в полном смысле этого слова.

Для создания базы данных в Django нужно:

- Создать аккаунт суперпользователя (админа)
- Написать модели, определяющие нужную структуру БД.
- Подготовить и выполнить миграции — «makemigrations» и «migrate».

Регистрация модели (в admin.py):

```
from .models import Info  
admin.site.register(Info)
```



```
20 from django.db import models  
21 class Info(models.Model):  
22     title = models.CharField('Заголовок', max_length=50)  
23     text = models.TextField('Объявление', max_length=300)  
24     date = models.DateTimeField('Время', max_length=30)  
25     def __str__(self):  
26         return self.title  
27     class Meta:  
28         verbose_name = 'Обновление'  
29         verbose_name_plural = 'Обновления'
```

Рисунок 8 — Создание модели

ОСНОВНАЯ ЧАСТЬ:

Вывод записей из базы данных

```
def info(request):  
    info = Info.objects.order_by('-date')  
    return render(request, "main/info.html", {'info' : info})
```

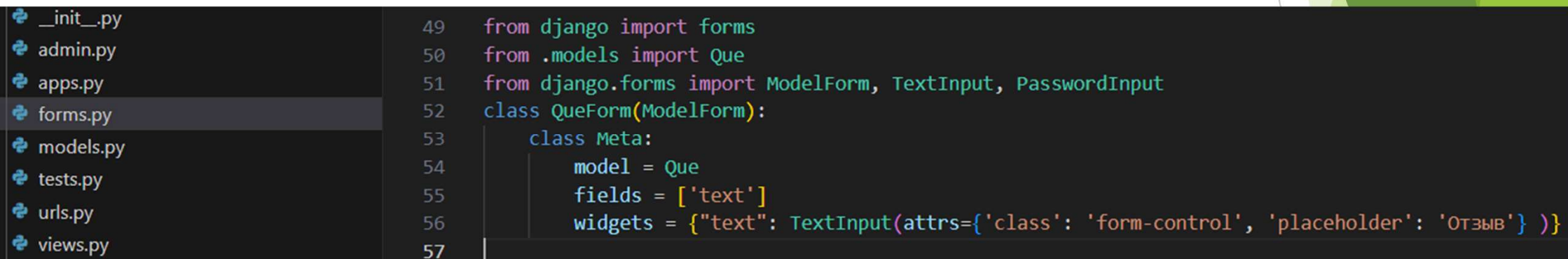
Рисунок 9 — Использование модели на странице

```
{% for el in info%}  
    <div class = "content">  
        <h3 class = "open"> {{ el.title }} </h3>  
        <p class = "anons" style = "font-weight:normal"> {{ el.text }} </p>  
        <p class = "data"> {{ el.date }} </p>  
    </div>  
{% endfor %}
```

Рисунок 10 — Вывод записей

ОСНОВНАЯ ЧАСТЬ: Создание и использование формы

После написания и регистрации модели напомним форму по заданной модели.



```
49 from django import forms
50 from .models import Que
51 from django.forms import ModelForm, TextInput, PasswordInput
52 class QueForm(ModelForm):
53     class Meta:
54         model = Que
55         fields = ['text']
56         widgets = {"text": TextInput(attrs={'class': 'form-control', 'placeholder': 'Отзыв'})}
57
```

Рисунок 11 — Структура формы

ОСНОВНАЯ ЧАСТЬ: Создание и использование формы

```
165 from .models import Que
166 from .forms import QueForm
167
168 def questions(request):
169     que = Que.objects.all
170     return render(request, "main/questions.html", {'que' : que})
171
172 def otz(request):
173     error = ''
174     if request.method == 'POST':
175         form = QueForm(request.POST)
176         if form.is_valid():
177             form.save()
178             return HttpResponseRedirect(reverse('questions'))
179         else:
180             error = ''
181     form = QueForm()
182     data = {
183         'form': form,
184         'error': error
185     }
186     return render(request, "main/otz.html", data)
```

Рисунок 12 — Запись формы в views.py

```
{% csrf_token %}
<div class = "fields">
    <div class = "f1">
        <div><p>Введите отзыв</p></div>
        <div class = "line"></div>
    </div>
    <div class = "field">{{form.text}}</br></div>
</div>
```

Рисунок 13 — Запись формы в views.py

Обязательно нужно использовать {% csrf_token %} для верной работы формы.

ОСНОВНАЯ ЧАСТЬ: Тестирование

Таблица 1. Функциональное тестирование

№	Назначение теста	Значения исходных данных	Ожидаемый результат	Реакция программы	Вывод
1	Проверка работы передвижения пользователя по страницам с использованием ссылок.	Нажатие на текст в виде ссылки.	Перемещение на нужную страницу сайта.	Открытие ожидаемой страницы.	Работа верная
2	Проверка вывода (создания и добавления).	Создание, заполнение модели и выведение из базы данных.	Действия с моделью будут работать исправно.	Созданная модель отображает нужную информацию на сайте.	Работа верная
3	Правильная работа заполнения формы, внешнее добавление записей в базу данных; пример: оставление отзыва.	Создание, заполнение модели и выведение из базы данных. Создание формы для заполнения.	Действия с моделью будут работать исправно. При заполнении формы запись отправится в базу данных и затем отобразится на другой странице сайта.	Форма заполнилась верно и отправилась в базу данных; Созданная модель отображает форму на сайте и отзыв человека.	Работа верная

ОСНОВНАЯ ЧАСТЬ: Тестирование

Таблица 2. Оценочное тестирование

№ Пользователя	Удобство использования	Визуальный стиль	Эксплуатация
«1»	8	9	7
«2»	8	9	7
«3»	7	8	6
«4»	8	8	7
«5»	9	7	8
«6»	8	9	7
«7»	9	9	8
«8»	7	10	9
«9»	9	10	8
«10»	8	10	7
Средняя оценка:	8,1	8,9	7,4

Итоговая оценка по всем баллам: 8,13

Комментарии и отзывы:

- «Интересный, познавательный сайт»
- «Выглядит красиво и лаконично, вызывает хороший эффект»
- «Только о половине всех мест что-то знаю, найду для себя что-нибудь новое»

ЗАКЛЮЧЕНИЕ

В результате работы был создан сайт, поддерживающий функции перемещения между страницами, возможностью оставить отзыв или подробно ознакомиться с достопримечательностями нашей страны и быть в курсе последних новостей связанных с сферой экотуризма.

Итоговый функционал доступный для пользователя: ознакомление с информацией о достопримечательностях России, их описанием с визуальными элементами; чтение новостей о развитии сферы экотуризма; возможность оставлять отзывы на сайте.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Силин, П. А. "Проектирование и разработка веб-приложений." М.: Издательство, 2020.
2. Баранов, С. В. "Основы работы с Django." М.: Издательство, 2021.
3. Степанов, И. А. "Методы тестирования программного обеспечения." М.: Издательство, 2017.
4. Шабанов, Д. Ю. "Дизайн пользовательского интерфейса." М.: Издательство, 2020.
5. Нейросеть GigaChat