

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Московский политехнический университет»

Кафедра «Инфокогнитивные технологии»
Образовательная программа «Веб-технологии»

Отчет по курсовому проекту
по дисциплине «Основы инженерного проектирования»

Тема: «Электронная регистратура Рязанской области»

Выполнил:

Студент группы 191-321

Огородников Н. А.

подпись, дата

Принял:

Старший преподаватель

Данышина М.В.

подпись, дата

Москва 2021

ОГЛАВЛЕНИЕ

| | |
|--|----|
| Оглавление | 2 |
| Введение..... | 3 |
| Тип разработки | 3 |
| Задача..... | 3 |
| Аналоги | 3 |
| Основная часть | 4 |
| Анализ аналогов | 4 |
| Проектирование физической структуры..... | 6 |
| Создание django-приложения..... | 7 |
| Настройка административного интерфейса Django..... | 9 |
| Наполнение базы данных. | 13 |
| Реализация REST API | 14 |
| Реализация импорта и экспорта данных | 19 |
| Типовые запросы к базе данных | 20 |
| Заключение | 24 |
| Список используемых источников | 25 |

ВВЕДЕНИЕ

Тип разработки

Индивидуальный проект, направленный на демонстрацию и фиксацию навыков, полученных в течение 3-го семестра обучения. Разработка ведется поэтапно.

Задача

Спроектировать и создать реляционную базу данных для веб-сервиса анкетирования пользователей сайта и настроить административную панель для управления базой данных с помощью фреймворка Django. Административная панель поможет создавать анкеты и вопросы, контролировать их содержание и просматривать ответы пользователей.

Аналоги

- [Google Forms](#)
- [Survio](#)
- [EXAMINARE](#)
- [Simpoll](#)
- [Анкетолог](#)

ОСНОВНАЯ ЧАСТЬ

Анализ аналогов

Сервисы анкетирования широко распространены в наше время, поэтому у разрабатываемого решения есть множество аналогов. Вот некоторые из них:

- [Google Forms](#)
- [Survio](#)
- [EXAMINARE](#)
- [Simpoll](#)
- [Анкетолог](#)

Данные сервисы имеют схожую структуру, и процесс создания анкеты в них сводится к трем основным шагам:

1. Создание анкеты
2. Добавление секций анкеты (минимум 1)
3. Добавление вопросов (минимум 1)

Данные сервисы предоставляют широкие возможности по настройке анкет: добавление описаний, настройка типов вопросов, распределение на логические элементы (секции).

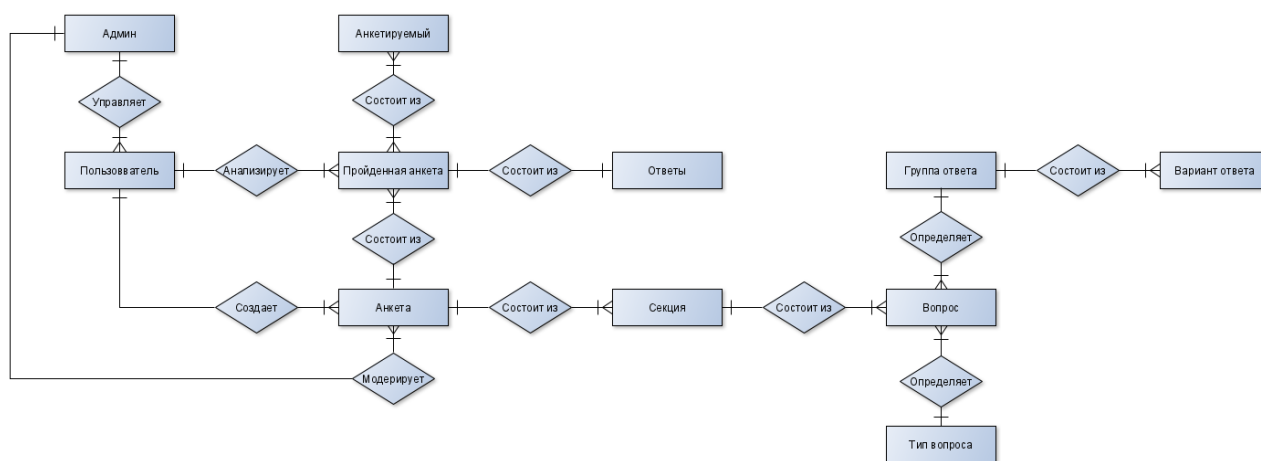
Все сервисы позволяют просматривать ответы, предоставленные пользователями, с целью дальнейшего анализа. Также есть поддержка экспорта ответов в большинстве современных форматов.

Опыт использования сервисов положительный. Интерфейс удобный и интуитивно понятный, присутствует адаптивность, что позволяет использовать сервисы на мобильных устройствах.

Подводя итог можно сказать, что сайты являются типовыми решениями, и имеют между собой высокую конкуренцию. Данные сервисы можно использовать как ориентир при разработке проекта.

Проектирование инфологической модели предметной области

Следуя из анализа конкурентов, сервис должен строиться на двух основных процессах: создание анкет и анализ ответов. Анкета состоит из секций, которые состоят из вопросов. Вопрос может быть определенного типа (один из, несколько из, ответ в свободной форме и так далее), также вопрос принадлежит к какой-либо группе ответа (в нее входят варианты ответа на вопрос). Группа ответов снижает количество строк в базе данных при создании анкет с вопросами, предполагающими одинаковые варианты ответов («согласен/не согласен/не знаю» и подобные).

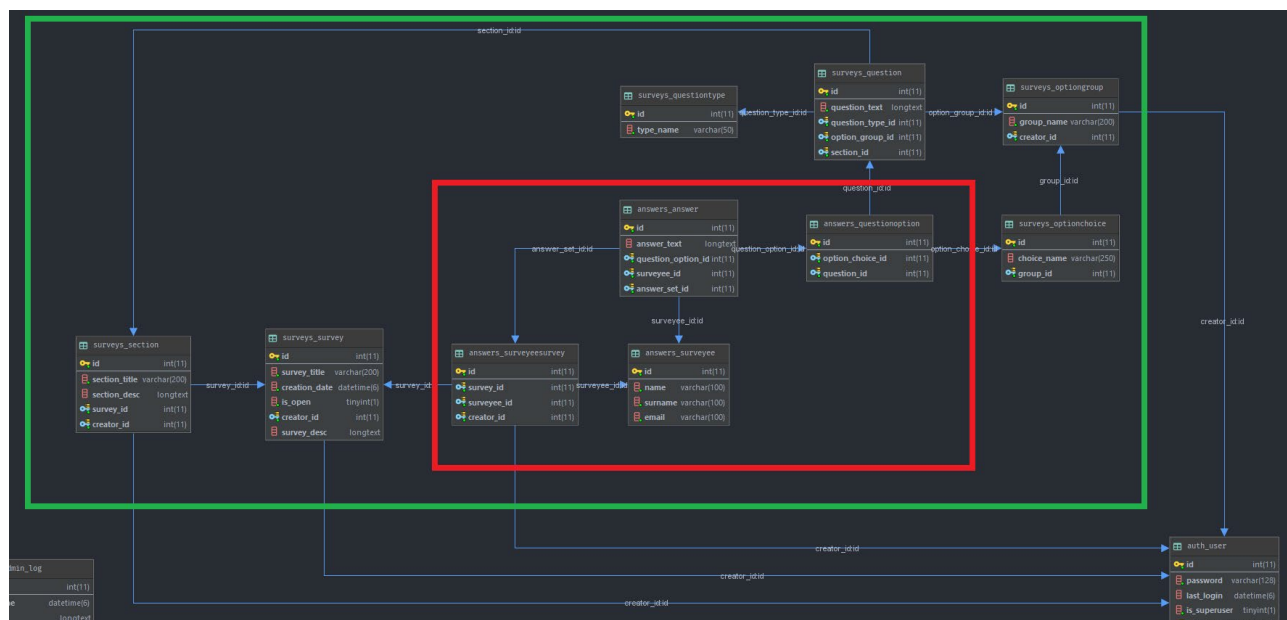


1. Инфологическая модель

Проектирование физической структуры

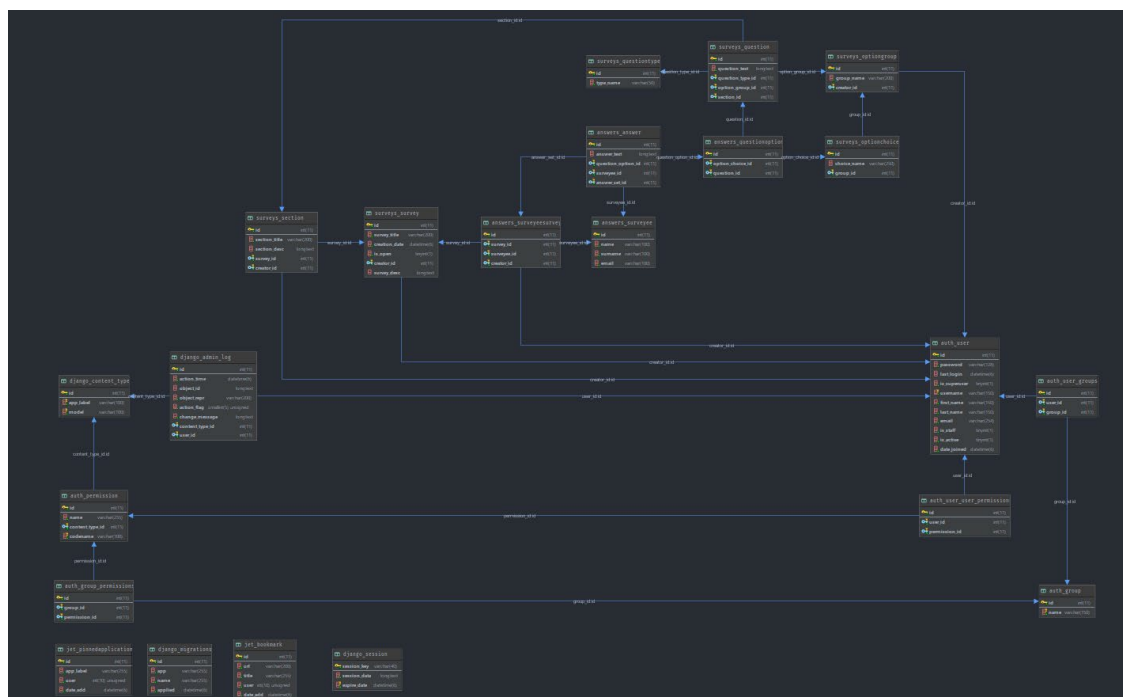
С помощью инфологической схемы можно составить подробную физическую схему базы данных. В ней находятся все те же сущности, но каждая из них имеет набор полей с определенным типом данных. Сущности связаны между собой по принципу, описанному в предыдущем пункте, однако имеется еще несколько связей с элементами базы данных фреймворка Django, обеспечивающие правильное функционирование системы.

Красным выделена часть базы данных, отвечающая за приложение ответов; зеленым – создания анкет.



2. Физическая структура

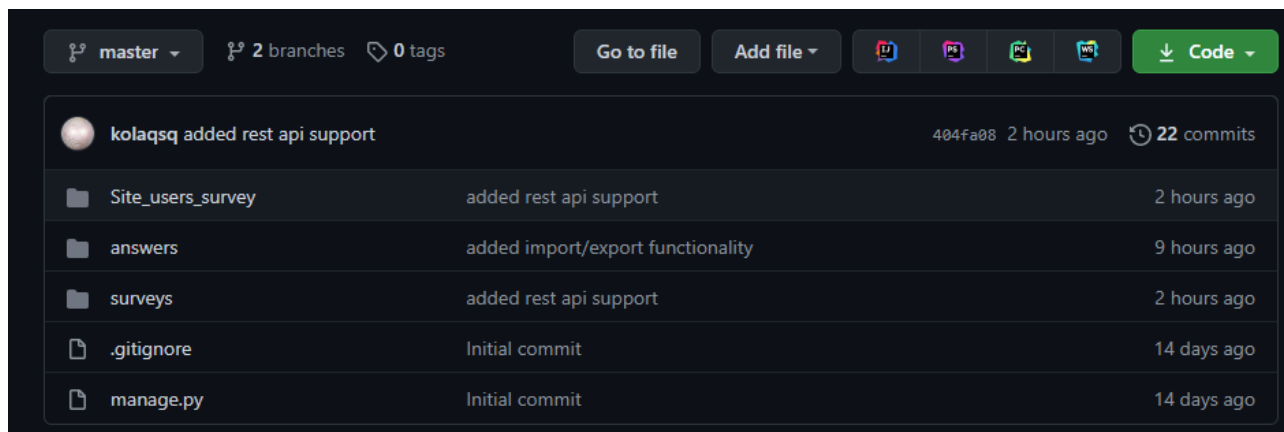
Полная схема базы данных, включающая таблицы фреймворка и его модулей выглядит следующим образом:



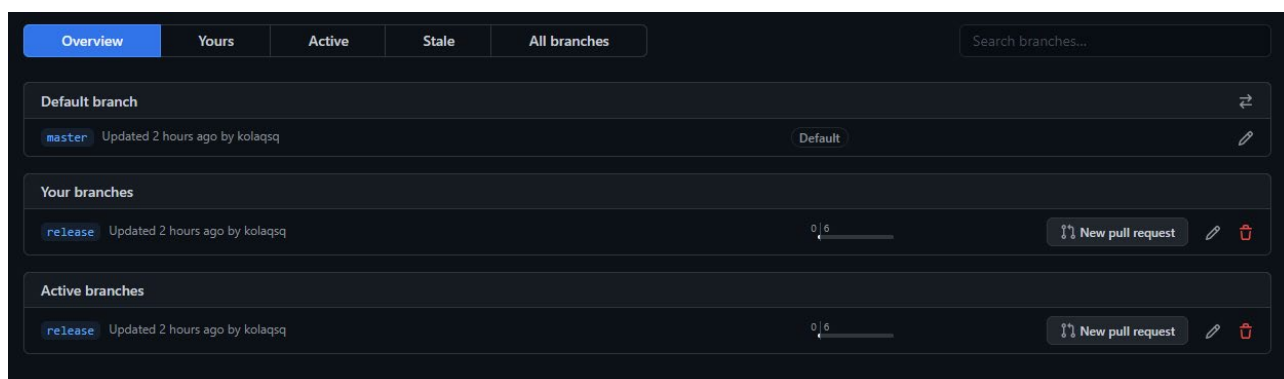
3. Полная схема БД

Создание django-приложения

После создания проект был загружен в GitHub в собственный репозиторий по адресу https://github.com/kolaqsk/Site_users_survey. В процессе разработки была добавлена ветка release, в которую загружалась версия приложения, настроенная под использования на веб-сервере fit.mospolytech.ru.

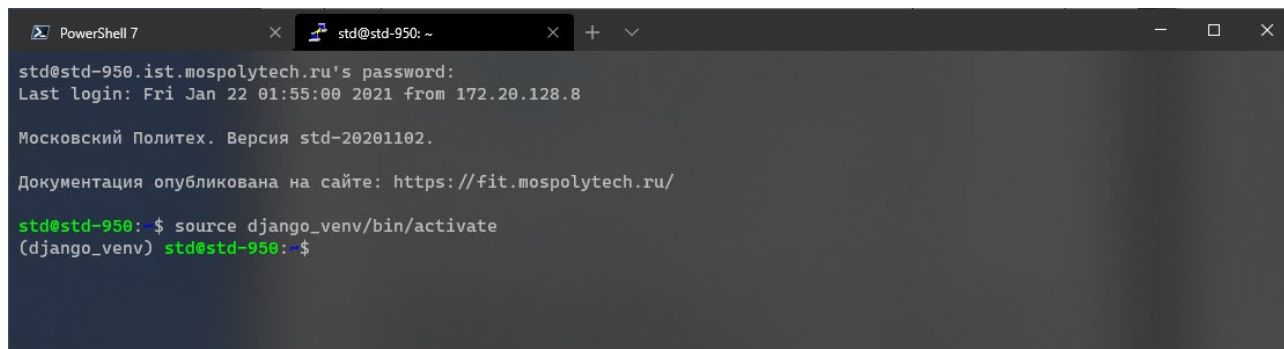


4. Git-репозиторий



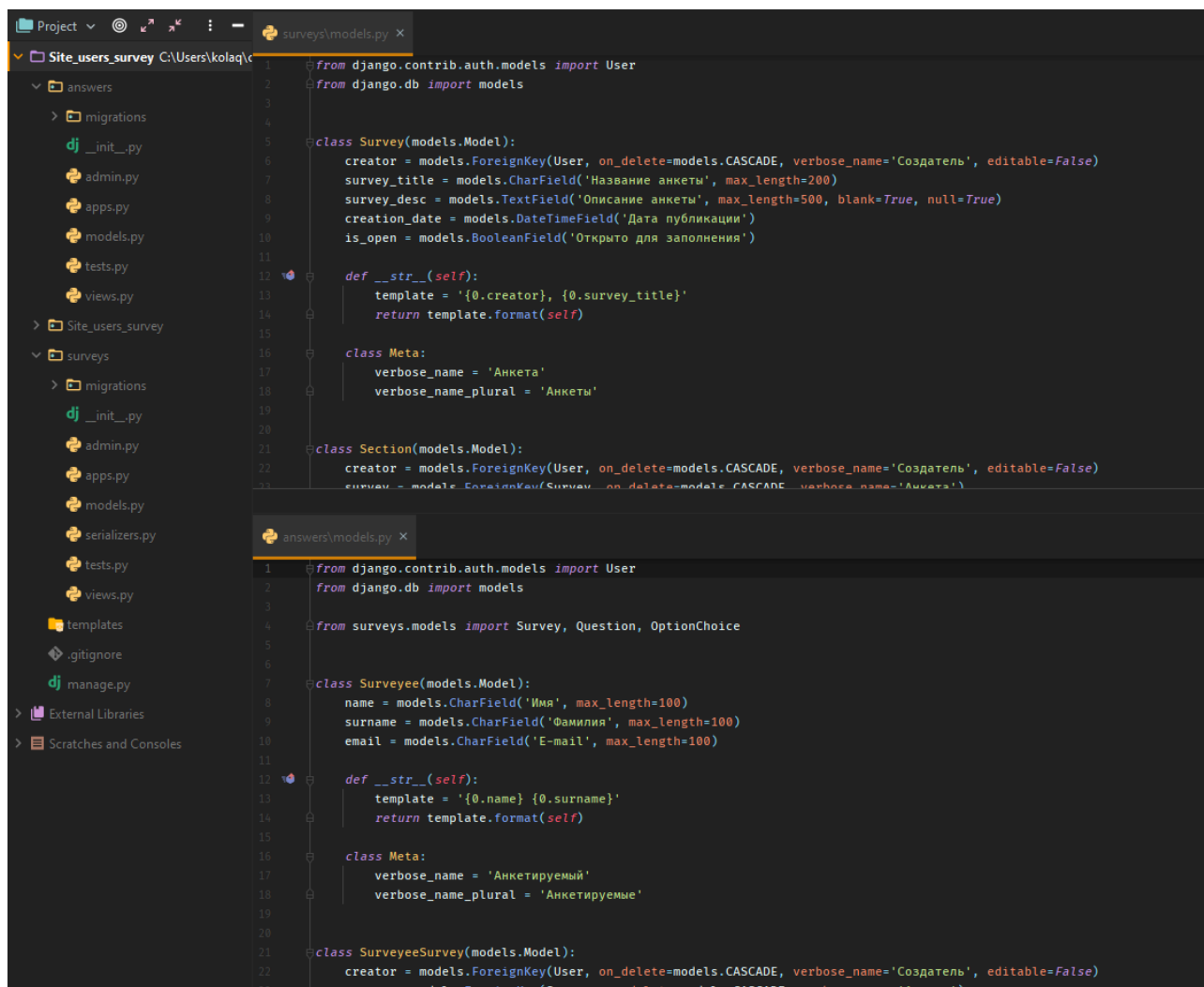
5. Ветки

На сервере было создано виртуальное python-окружение и установлены модули, необходимые для функционирования сервиса.



6. Виртуальное окружение

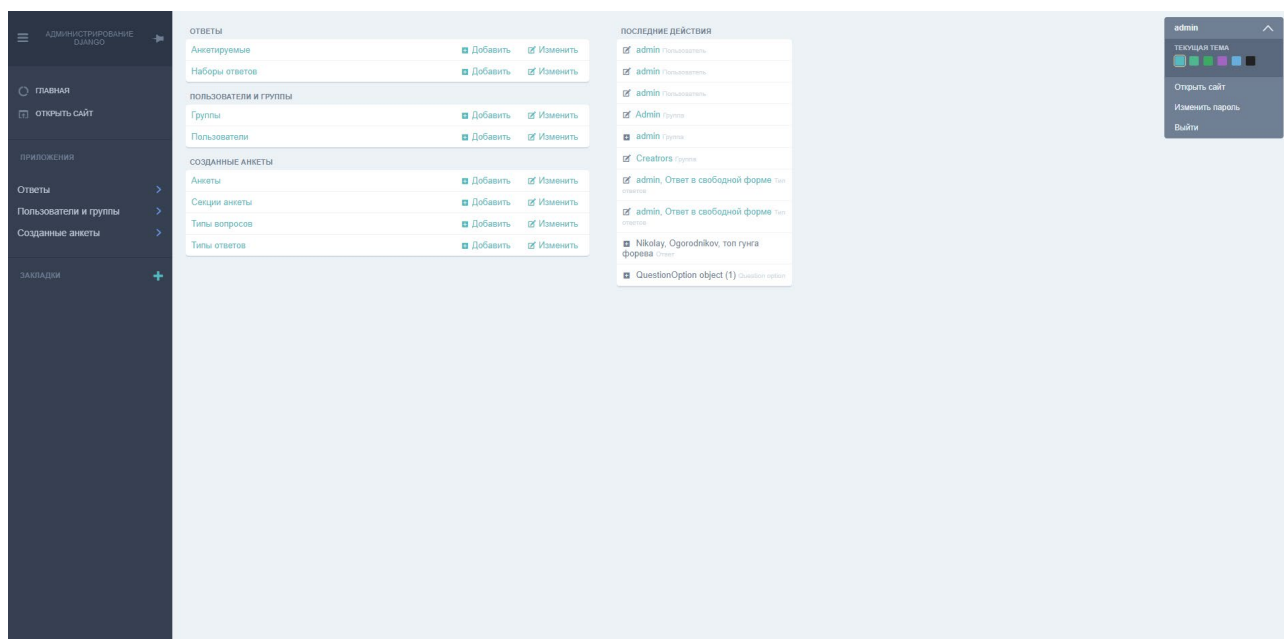
Далее в проекте было создано 2 приложения: surveys и answers. Первое нужно для создания и изменения анкет, а второе – для просмотра ответов анкетированных. В процессе создания приложений в них были перенесены таблицы в качестве моделей.



7. Структура проекта и models.py

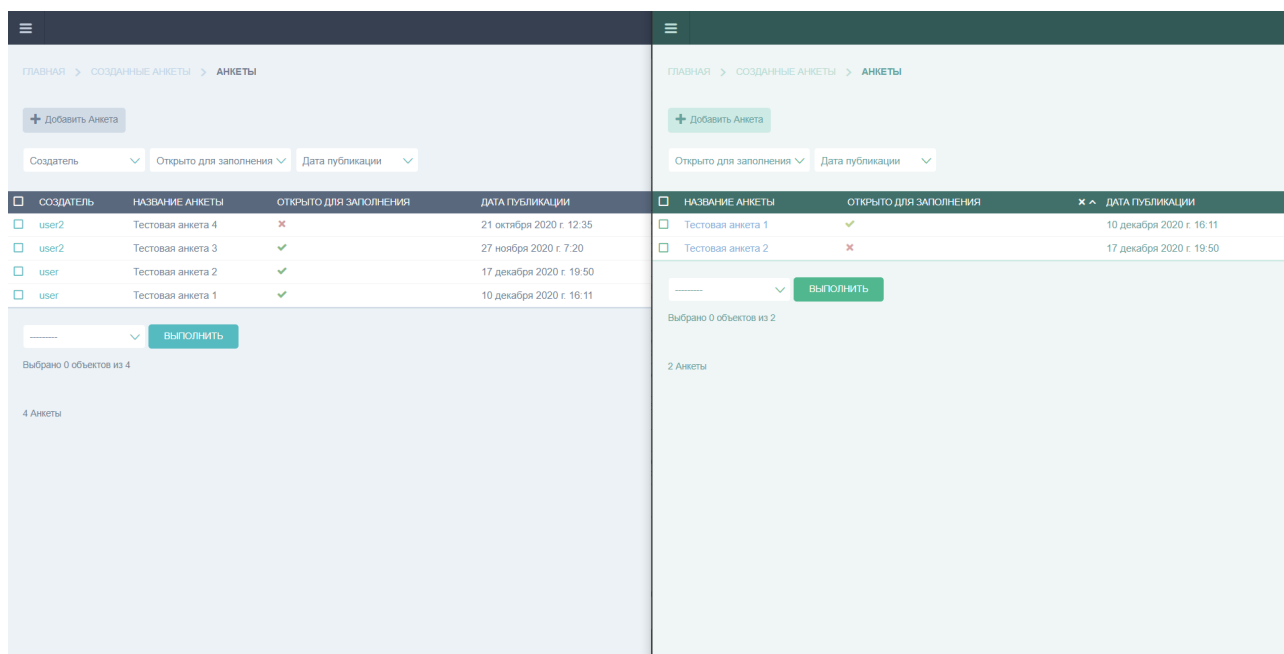
Настройка административного интерфейса Django

После создания приложения был установлен сторонний графический интерфейс Django JET, поскольку мне он показался более удобным.



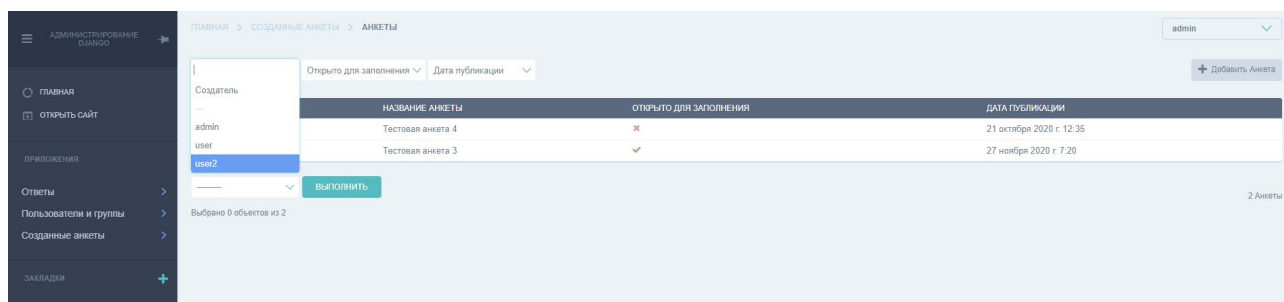
8. Интерфейс Django JET

Далее были созданы пользователи: admin/admin (суперпользователь), user/user и user2/user2 (обычные пользователи системы). Суперпользователь может видеть все сущности и редактировать их. Обычные пользователи могут создавать и редактировать сущности, принадлежащие этому пользователю, в приложении анкет. В приложении ответов обычный пользователь может только просматривать ответы на созданные им анкеты и список анketируемых людей. Они также могут удалять ответы.

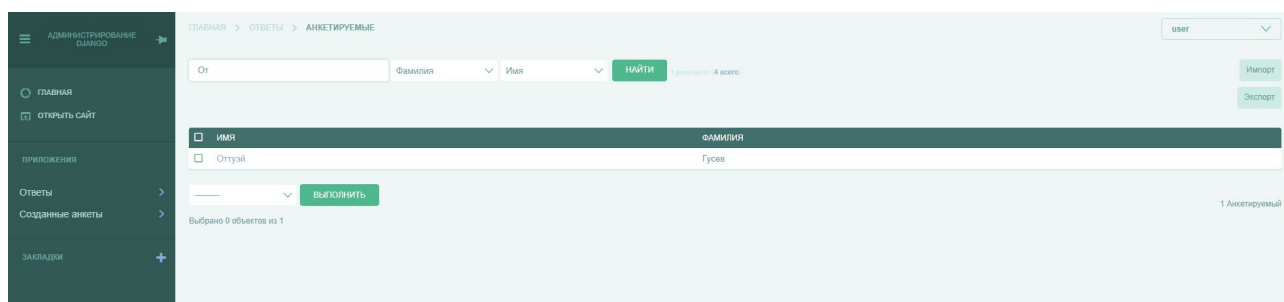


9. Отличия интерфейса у admin (слева) и user (справа)

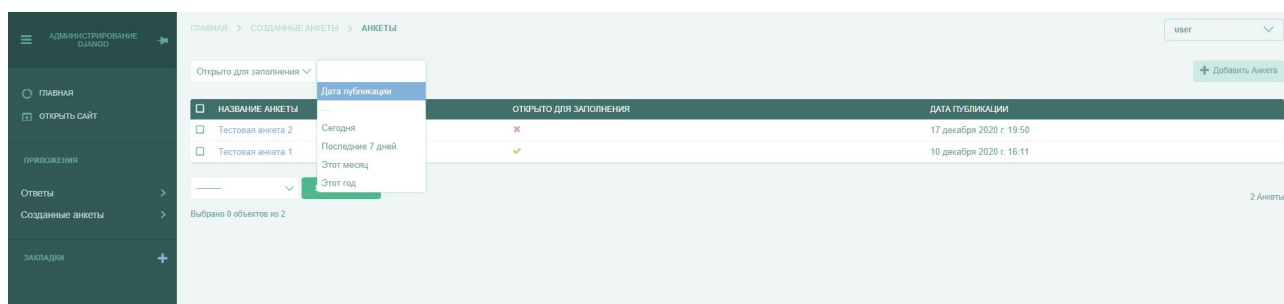
В процессе настройки интерфейса были применены такие методы как: фильтрация, поиск, фильтр по дате, сортировка, вывод информации из других таблиц, вложенность моделей.



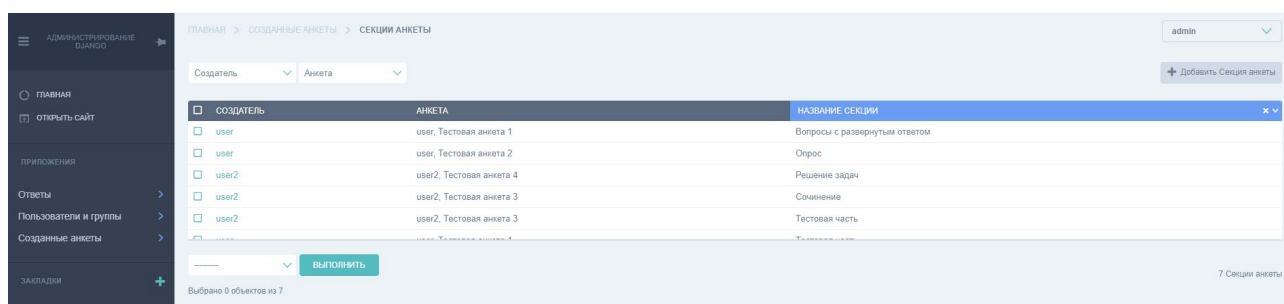
10. Фильтрация



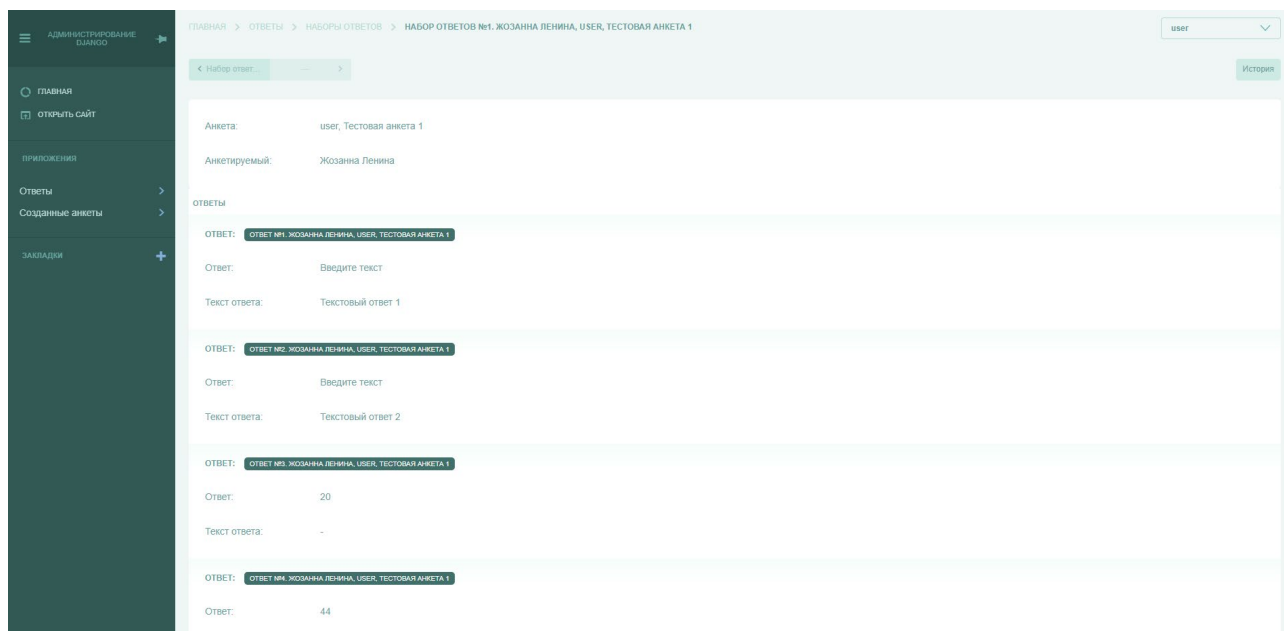
11. Поиск



12. Фильтр по дате



13. Сортировка



14. Вложенность

```
class SurveyeeSurveyAdmin(ImportExportModelAdmin):
    resource_class = SurveyeeSurveyResource
    inlines = [AnswerInline]

    def get_list_display(self, request):
        if request.user.is_superuser:
            return ['creator', 'survey_title', 'name', 'surname']
        else:
            return ['survey_title', 'name', 'surname']

    def get_list_filter(self, request):
        if request.user.is_superuser:
            return ['creator', 'survey']
        else:
            return ['survey']

    def get_queryset(self, request):
        query = SurveyeeSurvey.objects.filter(creator=request.user)
        if request.user.is_superuser:
            query = SurveyeeSurvey.objects.all()
        return query

    def survey_title(self, obj):
        return obj.survey.survey_title

    survey_title.short_description = 'Анкета'

    def name(self, obj):
        return obj.surveyee.name

    name.short_description = 'Имя'

    def surname(self, obj):
        return obj.surveyee.surname

    surname.short_description = 'Фамилия'
```

15. Вывод информации из других таблиц

Также в интерфейсе были имплементированы admin-actions. Были реализованы две функции, позволяющие открыть или закрыть для прохождения выбранные анкеты. При необходимости открыть или закрыть несколько анкет сразу, можно использовать admin-actions, а не редактировать каждую анкеты по отдельности.

Главная > СОЗДАНИЕ АНКЕТЫ > АНКЕТЫ

Создатель: [v] Открыто для заполнения: [v] Дата публикации: [v]

+ Добавить Анкету

| СОЗДАТЕЛЬ | НАЗВАНИЕ АНКЕТЫ | ОТКРЫТО ДЛЯ ЗАПОЛНЕНИЯ | ДАТА ПУБЛИКАЦИИ |
|-------------------------------------|-------------------|------------------------|--------------------------|
| Удалить выбранные Анкеты | Тестовая анкета 4 | ✗ | 21 октября 2020 г. 12:35 |
| Сделать доступными для заполнения | Тестовая анкета 3 | ✓ | 27 ноября 2020 г. 7:20 |
| Сделать недоступными для заполнения | Тестовая анкета 2 | ✗ | 17 декабря 2020 г. 19:50 |
| | Тестовая анкета 1 | ✓ | 10 декабря 2020 г. 16:11 |

Выполнить

Выбрано 2 из 4

4 Анкеты

16. Admin-actions

```

7
8 def make_open(modeladmin, request, queryset):
9     queryset.update(is_open=True)
10
11
12     make_open.short_description = 'Сделать доступными для заполнения'
13
14
15 def make_close(modeladmin, request, queryset):
16     queryset.update(is_open=False)
17
18
19     make_close.short_description = 'Сделать недоступными для заполнения'
20

```

17. Код для admin-actions

Наполнение базы данных.

На данном этапе база данных разрабатываемого сервиса была заполнена тестовыми данными.

| | id | survey_title | creation_date | is_open | creator_id | survey_desc |
|---|----|-------------------|----------------------|---------|------------|--|
| 1 | 47 | Тестовая анкета 1 | 2020-12-10 13:11:... | 1 | 2 | Тестовая анкета №1 созданная пользователем User |
| 2 | 48 | Тестовая анкета 2 | 2020-12-17 16:50:... | 0 | 2 | Тестовая анкета №2 созданная пользователем User |
| 3 | 49 | Тестовая анкета 3 | 2020-11-27 04:20:... | 1 | 3 | Тестовая анкета №3 созданная пользователем User2 |
| 4 | 50 | Тестовая анкета 4 | 2020-10-21 09:35:... | 0 | 3 | Тестовая анкета №4 созданная пользователем User2 |

18. Таблица анкет

| | id | question_text | question_type_id | option_group_id | section_id |
|----|----|--|------------------|-----------------|------------|
| 1 | 6 | Тестовый вопрос 1 | 1 | 1 | 56 |
| 2 | 7 | Тестовый вопрос 2 | 1 | 1 | 56 |
| 3 | 8 | Тестовый вопрос 3 | 2 | 7 | 57 |
| 4 | 9 | Согласны ли вы с данным утверждением 1 | 2 | 3 | 58 |
| 5 | 10 | Согласны ли вы с данным утверждением 2 | 2 | 3 | 58 |
| 6 | 11 | Соответствует ли данное утверждение вашему характеру? | 2 | 4 | 58 |
| 7 | 12 | Тестовый вопрос 1 | 2 | 8 | 59 |
| 8 | 13 | Тестовый вопрос 2 | 2 | 9 | 59 |
| 9 | 14 | Тестовый вопрос 4 | 2 | 10 | 57 |
| 10 | 15 | Почему надо делать так, а не эдак? | 1 | 1 | 60 |
| 11 | 16 | Что значит так? А что эдак? | 1 | 1 | 60 |
| 12 | 17 | Напишите сочинение на тему "Настройка админки django" | 1 | 1 | 61 |
| 13 | 18 | Решите задачу. На размышление 30 черных точек | 1 | 1 | 62 |
| 14 | 19 | Висит груша нельзя скушать. Кто это? Решение должно занимать ми... | 1 | 1 | 62 |

19. Таблица вопросов

Также были разработаны кейсы использования приложения пользователем:

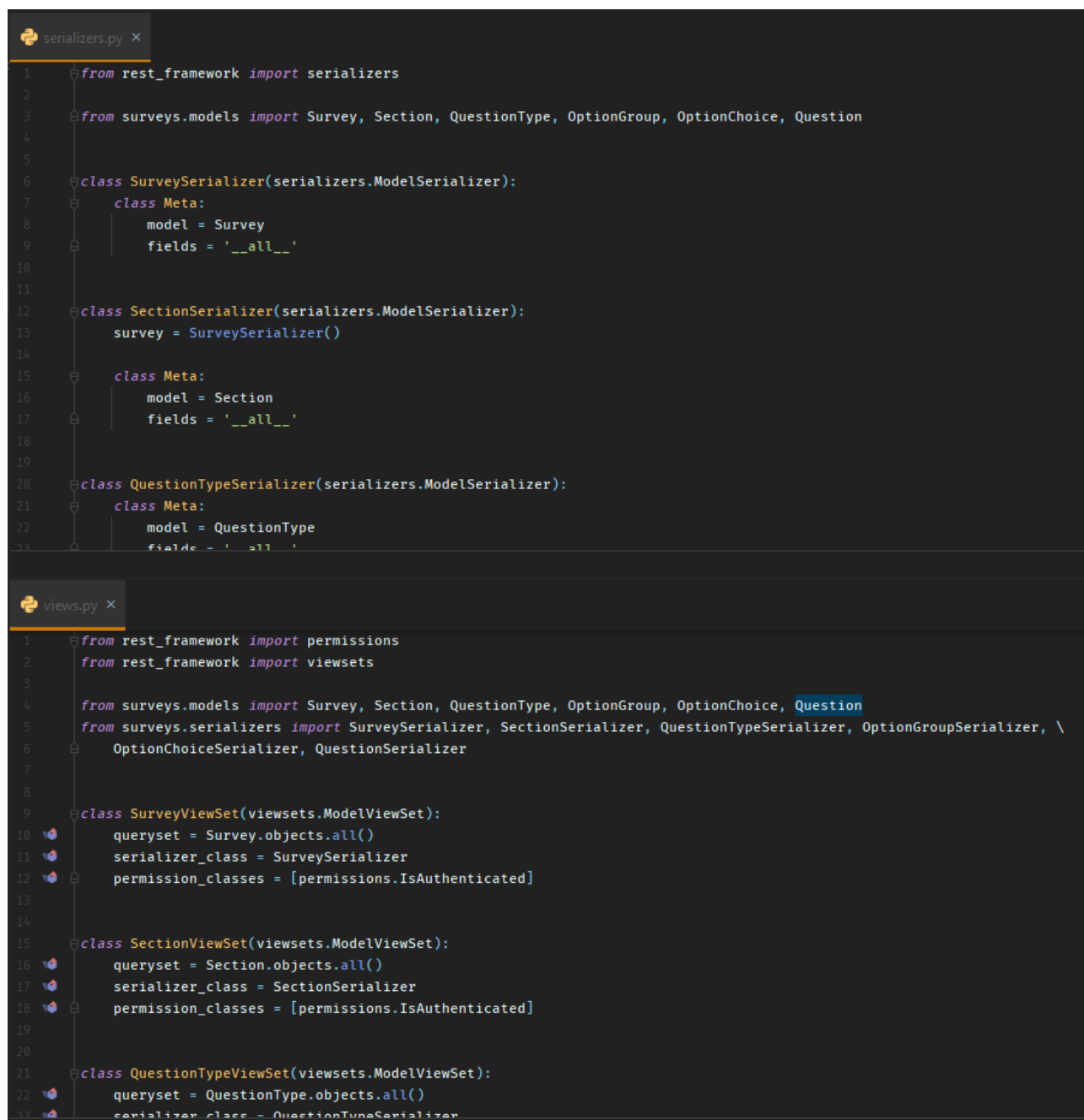
- Редактирование данных пользователя
- Удаление сущностей в приложении анкет
- Удаление сущностей в приложении ответов
- Добавление сущностей в приложении анкет
- Редактирование сущностей в приложении анкет
- Просмотр сущностей в приложении ответов

и администратором:

- Редактирование прав пользователя
- Создание групп пользователей с определенными правами
- Редактирование групп пользователей с определенными правами
- Редактирование данных пользователя
- Редактирование списка типов вопросов
- Удаление сущностей в приложении анкет, если они не удовлетворяют требованиям сервиса (неприемлемое содержание и т.п.)
- Удаление сущностей в приложении ответов, если они не удовлетворяют требованиям сервиса (неприемлемое содержание и т.п.)
- Добавление сущностей в приложении анкет
- Добавление сущностей в приложении ответов
- Редактирование сущностей в приложении анкет
- Редактирование сущностей в приложении ответов

Реализация REST API

На данном этапе был установлен Django REST API, создан и настроен `serializer.py`, добавлены URL.



The image shows a code editor with two files open: `serializers.py` and `views.py`.

serializers.py

```

1  from rest_framework import serializers
2
3  from surveys.models import Survey, Section, QuestionType, OptionGroup, OptionChoice, Question
4
5
6  class SurveySerializer(serializers.ModelSerializer):
7      class Meta:
8          model = Survey
9          fields = '__all__'
10
11
12  class SectionSerializer(serializers.ModelSerializer):
13      survey = SurveySerializer()
14
15      class Meta:
16          model = Section
17          fields = '__all__'
18
19
20  class QuestionTypeSerializer(serializers.ModelSerializer):
21      class Meta:
22          model = QuestionType
23          fields = '__all__'

```

views.py

```

1  from rest_framework import permissions
2  from rest_framework import viewsets
3
4  from surveys.models import Survey, Section, QuestionType, OptionGroup, OptionChoice, Question
5  from surveys.serializers import SurveySerializer, SectionSerializer, QuestionTypeSerializer, OptionGroupSerializer, \
6      OptionChoiceSerializer, QuestionSerializer
7
8
9  class SurveyViewSet(viewsets.ModelViewSet):
10     queryset = Survey.objects.all()
11     serializer_class = SurveySerializer
12     permission_classes = [permissions.IsAuthenticated]
13
14
15  class SectionViewSet(viewsets.ModelViewSet):
16     queryset = Section.objects.all()
17     serializer_class = SectionSerializer
18     permission_classes = [permissions.IsAuthenticated]
19
20
21  class QuestionTypeViewSet(viewsets.ModelViewSet):
22     queryset = QuestionType.objects.all()
23     serializer_class = QuestionTypeSerializer

```

20. `serializer.py` u `views.py`

```

16 from django.conf.urls import url
17 from django.contrib import admin
18 from django.urls import path, include
19 from rest_framework import routers
20
21 from surveys import views
22
23 router = routers.DefaultRouter()
24 router.register(r'survey', views.SurveyViewSet)
25 router.register(r'section', views.SectionViewSet)
26 router.register(r'question_type', views.QuestionTypeViewSet)
27 router.register(r'option_group', views.OptionGroupViewSet)
28 router.register(r'option_choice', views.OptionChoiceViewSet)
29 router.register(r'question', views.QuestionViewSet)
30
31 urlpatterns = [
32     url(r'^jet/', include('jet.urls', 'jet')),
33     path('admin/', admin.site.urls),
34     path('', include(router.urls)),
35     path('api-auth/', include('rest_framework.urls', namespace='rest_framework'))
36 ]

```

21. *urls.py*

В итоге получились страницы следующего типа. На них отображается GET запрос по соответствующему serializer-y.

The screenshot shows a web browser interface for a Django REST framework API. The top navigation bar includes 'Django REST framework' and a user profile 'admin'. The breadcrumb trail is 'Api Root / Question List'. The main heading is 'Question List', with 'OPTIONS' and 'GET' buttons to its right. Below the heading is a pagination control showing '1' of 2 pages. The main content area displays the details of a GET request to '/question/'. The response is a 200 OK status with headers: 'Allow: GET, POST, HEAD, OPTIONS', 'Content-Type: application/json', and 'Vary: Accept'. The response body is a JSON object:

```

{
  "count": 14,
  "next": "http://site-users-survey.std-950.ist.mospolytech.ru/question/?page=2",
  "previous": null,
  "results": [
    {
      "id": 6,
      "section": {
        "id": 56,
        "survey": {
          "id": 47,
          "survey_title": "Тестовая анкета 1",
          "survey_desc": "Тестовая анкета №1 созданная пользователем User",
          "creation_date": "2020-12-10T16:11:00+03:00",
          "is_open": true,
          "creator": 2
        },
        "section_title": "Вопросы с развернутым ответом",
        "section_desc": "В данной секции анкетиреуемому необходимо предоставить развернутый ответ на",
        "creator": 2
      },
      "question_type": {
        "id": 1,
        "type_name": "Текст"
      },
      "option_group": {
        "id": 1,
        "group_name": "Ответ в свободной форме",
        "creator": 1
      },
      "question_text": "Тестовый вопрос 1"
    }
  ]
}

```

22. Страница REST API

Далее API было протестировано в программе Postman.

The screenshot shows the Postman interface with a GET request to `http://site-users-survey.std-950.ist.mospolytech.ru/survey`. The response is a JSON array of four survey objects, displayed in the 'Body' tab with a 'Pretty' view. The status is 200 OK, 237 ms, 1.28 KB.

```

[
  {
    "survey_desc": "Тестовая анкета №1 созданная пользователем User",
    "creation_date": "2020-12-10T16:11:00+03:00",
    "is_open": true,
    "creator": 2
  },
  {
    "id": 48,
    "survey_title": "Тестовая анкета 2",
    "survey_desc": "Тестовая анкета №2 созданная пользователем User",
    "creation_date": "2020-12-17T19:50:00+03:00",
    "is_open": false,
    "creator": 2
  },
  {
    "id": 49,
    "survey_title": "Тестовая анкета 3",
    "survey_desc": "Тестовая анкета №3 созданная пользователем User2",
    "creation_date": "2020-11-27T07:20:00+03:00",
    "is_open": true,
    "creator": 3
  },
  {
    "id": 50,
    "survey_title": "Тестовая анкета 4",
    "survey_desc": "Тестовая анкета №4 созданная пользователем User2",
    "creation_date": "2020-10-21T12:35:00+03:00",
    "is_open": false,
    "creator": 3
  }
]

```

23. Запрос перед добавлением

POST http://site-user... GET http://site-user... POST http://site-user... + ... No Environment

Untitled Request BUILD

POST http://site-users-survey.std-950.ist.mospolytech.ru/survey Send Save

Params Authorization Headers (9) Body Pre-request Script Tests Settings Cookies Code

none form-data x-www-form-urlencoded raw binary GraphQL Text

```

1 {
2   "id": 52,
3   "survey_title": "Тестовая анкета Postman",
4   "survey_desc": "Тестовая анкета Postman созданная пользователем admin",
5   "creation_date": "2021-01-17T19:50:00+03:00",
6   "is_open": true,
7   "creator": 1
8 }

```

Body Cookies Headers (11) Test Results 200 OK 231 ms 1.28 KB Save Response

Pretty Raw Preview Visualize JSON

```

18   "creation_date": "2020-12-17T19:50:00+03:00",
19   "is_open": false,
20   "creator": 2
21 },
22 {
23   "id": 49,
24   "survey_title": "Тестовая анкета 3",
25   "survey_desc": "Тестовая анкета №3 созданная пользователем User2",
26   "creation_date": "2020-11-27T07:20:00+03:00",
27   "is_open": true,
28   "creator": 3
29 },
30 {
31   "id": 50,
32   "survey_title": "Тестовая анкета 4",
33   "survey_desc": "Тестовая анкета №4 созданная пользователем User2",
34   "creation_date": "2020-10-21T12:35:00+03:00",
35   "is_open": false,
36   "creator": 3
37 }
38 ]
39 }

```

24. Запрос-добавление

POST http://site-user... GET http://site-user... X POST http://site-user... + ... No Environment

Untitled Request BUILD

GET http://site-users-survey.std-950.ist.mospolytech.ru/survey Send Save

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies Code

Query Params

| KEY | VALUE | DESCRIPTION | ... | Bulk Edit |
|-----|-------|-------------|-----|-----------|
| Key | Value | Description | | |

Body Cookies Headers (11) Test Results 200 OK 224 ms 1.52 KB Save Response

Pretty Raw Preview Visualize JSON

```

26     "creation_date": "2020-11-27T07:20:00+03:00",
27     "is_open": true,
28     "creator": 3
29   },
30   {
31     "id": 50,
32     "survey_title": "Тестовая анкета 4",
33     "survey_desc": "Тестовая анкета №4 созданная пользователем User2",
34     "creation_date": "2020-10-21T12:35:00+03:00",
35     "is_open": false,
36     "creator": 3
37   },
38   {
39     "id": 52,
40     "survey_title": "Тестовая анкета Postman",
41     "survey_desc": "Тестовая анкета Postman созданная пользователем admin",
42     "creation_date": "2021-01-17T19:50:00+03:00",
43     "is_open": true,
44     "creator": 1
45   }
46 ]
47 }

```

25. Запрос после добавления

главная > СОЗДАНИЕ АНКЕТЫ > АНКЕТЫ admin

Создатель Открыто для заполнения Дата публикации

+ Добавить Анкету

| СОЗДАТЕЛЬ | НАЗВАНИЕ АНКЕТЫ | ОТКРЫТО ДЛЯ ЗАПОЛНЕНИЯ | ДАТА ПУБЛИКАЦИИ |
|--------------------------------|-------------------------|------------------------|--------------------------|
| <input type="checkbox"/> admin | Тестовая анкета Postman | ✓ | 17 января 2021 г. 19:50 |
| <input type="checkbox"/> user2 | Тестовая анкета 4 | ✗ | 21 октября 2020 г. 12:35 |
| <input type="checkbox"/> user2 | Тестовая анкета 3 | ✓ | 27 ноября 2020 г. 7:20 |
| <input type="checkbox"/> user1 | Тестовая анкета 2 | ✗ | 17 декабря 2020 г. 19:50 |
| <input type="checkbox"/> user1 | Тестовая анкета 1 | ✓ | 10 декабря 2020 г. 16:11 |

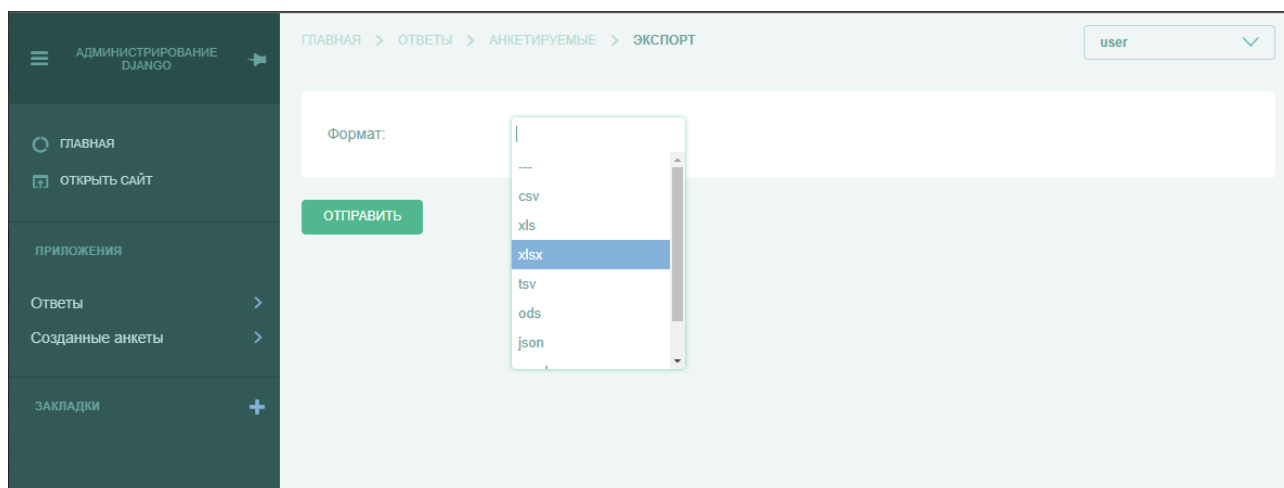
ВЫПОЛНИТЬ

Выбрано 0 объектов из 5 5 Анкеты

26. Админ-панель после добавления

Реализация импорта и экспорта данных

С помощью django-import-export были реализованы возможности импорта и экспорта данных.



27. Страница экспорта

The screenshot shows an Excel spreadsheet titled 'Surveyee-2021-01-22.xlsx'. The data is organized in a table with the following columns: id, name, surname, and email. The data rows are as follows:

| | id | name | surname | email |
|----|----|-----------|------------|-------------------------------|
| 1 | 4 | Златовлад | Круглов | ZlatovladKruglov31@mail.com |
| 2 | 3 | Оттуэй | Гусев | OttueyGusev858@mail.com |
| 3 | 2 | Аннетте | Павловская | AnnettePavlovskaya97@mail.com |
| 4 | 1 | Жозанна | Ленина | ZhozannaLenina403@mail.com |
| 5 | | | | |
| 6 | | | | |
| 7 | | | | |
| 8 | | | | |
| 9 | | | | |
| 10 | | | | |

The bottom of the Excel window shows a 'Tablib Dataset' tab and a zoom level of 100%.

28. Экспортированные данные в формате .xlsx

Типовые запросы к базе данных

Далее представлены типовые запросы к базе данных и соответствующие им представления в панели администратора.

```
1 select ss.creator_id, survey_title, name, surname
2 from answers_surveyee
3      join answers_surveyeesurvey `as` on answers_surveyee.id = `as`.surveyee_id
4      join surveys_survey ss on `as`.survey_id = ss.id
5 where ss.creator_id = 2
```

29. Типовой запрос 1

| АНКЕТА | ИМЯ | ФАМИЛИЯ |
|-------------------|---------|------------|
| Тестовая анкета 2 | Аннетте | Павловская |
| Тестовая анкета 2 | Жозанна | Ленина |
| Тестовая анкета 1 | Жозанна | Ленина |

30. Представление типового запроса 1

```
1 select survey_title, is_open, creation_date
2 from surveys_survey
3 where creator_id = 2
4 and is_open = 1
```

31. Типовой запрос 2

| НАЗВАНИЕ АНКЕТЫ | ОТКРЫТО ДЛЯ ЗАПОЛНЕНИЯ | ДАТА ПУБЛИКАЦИИ |
|-------------------|------------------------|--------------------------|
| Тестовая анкета 1 | ✓ | 10 декабря 2020 г. 16:11 |

32. Представление типового запроса 2

```

1  ✓ select survey_title, survey_desc, creation_date, is_open, section_title, section_desc
2    from surveys_survey ssu
3      join surveys_section ss on ssu.id = ss.survey_id
4    where ssu.creator_id = 2
5    and ssu.id = 47

```

33. Типовой запрос 3

ADMINISTRATOR: user

Путь: ГЛАВНАЯ > СОЗДАНИЕ АНКЕТЫ > АНКЕТЫ > USER, ТЕСТОВАЯ АНКЕТА 1

Название анкеты: * Тестовая анкета 1

Описание анкеты: Тестовая анкета №1 созданная пользователем User

Дата публикации: * 10.12.2020 16:11:00

☒ Открыто для заполнения

СЕКЦИИ АНКЕТЫ

СЕКЦИЯ АНКЕТЫ: USER, ТЕСТОВАЯ АНКЕТА 1, ВОПРОСЫ С РАЗВЕРНУТЫМ ОТВЕТОМ УДАЛЕНО

Название секции: * Вопросы с развернутым ответом

Описание секции: В данной секции анкетированному необходимо предоставить развернутый ответ на вопрос

СЕКЦИЯ АНКЕТЫ: USER, ТЕСТОВАЯ АНКЕТА 1, ТЕСТОВАЯ ЧАСТЬ УДАЛЕНО

Название секции: * Тестовая часть

Описание секции: В данной секции анкетированному необходимо ответить на тестовые вопросы

34. Представление типового запроса 3

```

1  ✓ select survey_title, section_title, section_desc, q.id, question_type_id, option_group_id, question_text
2    from surveys_question q
3      join surveys_section ss on ss.id = q.section_id
4      join surveys_survey s on ss.survey_id = s.id
5    where ss.id = 57
6

```

35. Типовой запрос 4

Анкета:* user, Тестовая анкета 1

Название секции:* Тестовая часть

Описание секции: В данной секции анкетиреваемому необходимо ответить на тестовые вопросы

ВОПРОСЫ

ВОПРОС: ВОПРОС №8. ТЕСТОВАЯ ЧАСТЬ. USER, ТЕСТОВАЯ АНКЕТА 1 УДАЛЕНО

Тип вопроса:* Один из

Тип ответа:* user, Ответ для тестового вопроса 3

Вопрос:* Тестовый вопрос 3

ВОПРОС: ВОПРОС №14. ТЕСТОВАЯ ЧАСТЬ. USER, ТЕСТОВАЯ АНКЕТА 1 УДАЛЕНО

Тип вопроса:* Один из

Тип ответа:* user, Ответ для тестового вопроса 4

36. Представление типового запроса 4

```

1  select as2.id,
2      s.id,
3      survey_title,
4      name,
5      surname,
6      a.id,
7      choice_name,
8      answer_text
9  from answers_answer a
10     join answers_questionoption aq on aq.id = a.question_option_id
11     join surveys_optionchoice so on so.id = aq.option_choice_id
12     join answers_surveyee `as` on `as`.id = a.surveyee_id
13     join surveys_question sq on sq.id = aq.question_id
14     join surveys_section ss on ss.id = sq.section_id
15     join surveys_survey s on s.id = ss.survey_id
16     join answers_surveyeesurvey as2 on `as`.id = as2.surveyee_id
17  where as2.id = 3
18  and s.id = 48

```

37. Типовой запрос 5

The screenshot displays a Django administration interface. On the left is a dark sidebar with navigation links: 'ADMINISTRИРОВАНИЕ DJANGO', 'ГЛАВНАЯ', 'ОТКРЫТЬ САЙТ', 'ПРИЛОЖЕНИЯ', 'Ответы', 'Созданные анкеты', and 'ЗАКЛАДКИ'. The main content area shows a breadcrumb trail: 'ГЛАВНАЯ > ОТВЕТЫ > НАБОРЫ ОТВЕТОВ > НАБОР ОТВЕТОВ №3. АННЕТТЕ ПАВЛОВ...'. A dropdown menu at the top right shows 'user'. Below the breadcrumb is a navigation bar with 'Набор ответ...' and 'История' buttons. The main content area lists survey responses for 'user, Тестовая анкета 2' by 'Анnette Павловская'. It shows three responses with their answers and text. At the bottom, there is a 'Закреть' button and a red 'x УДАЛЕНО' button.

ADMINISTRИРОВАНИЕ DJANGO

ГЛАВНАЯ > ОТВЕТЫ > НАБОРЫ ОТВЕТОВ > НАБОР ОТВЕТОВ №3. АННЕТТЕ ПАВЛОВ...

user

Набор ответ... История

Анкета: user, Тестовая анкета 2

Анketируемый: Анnette Павловская

ОТВЕТЫ

ОТВЕТ: ОТВЕТ №8. АННЕТТЕ ПАВЛОВСКАЯ, USER, ТЕСТОВАЯ АНКЕТА 2

Ответ: Не согласен

Текст ответа: -

ОТВЕТ: ОТВЕТ №9. АННЕТТЕ ПАВЛОВСКАЯ, USER, ТЕСТОВАЯ АНКЕТА 2

Ответ: Согласен

Текст ответа: -

ОТВЕТ: ОТВЕТ №10. АННЕТТЕ ПАВЛОВСКАЯ, USER, ТЕСТОВАЯ АНКЕТА 2

Ответ: Частично не соответствует

Текст ответа: -

Закреть x УДАЛЕНО

38. Представление типового запроса 5

ЗАКЛЮЧЕНИЕ

В результате процесса разработки получилось то, что и изначально планировалось: приложение с настроенной административной панелью на фреймворке Django.

Также в процессе создания данного приложения я применил знания и навыки, полученные за прошлые курсы обучения (python-разработка, проектирование баз данных), освежил их и даже в некоторой степени развил.

Полноценная Full-Stack разработка была новым и интересным опытом в моей практике, и я намерен дальше совершенствовать свои навыки в данном направлении.

Git-репозиторий: https://github.com/kolaqsq/Site_users_survey

Приложение: <http://site-users-survey.std-950.ist.mospolytech.ru/admin/>

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

<https://docs.djangoproject.com/en/3.1/>

<https://webdevblog.ru/kak-prevratit-adminku-django-v-legkovesnuju-panel-instrumentov>

<https://djbook.ru/>

<https://django-import-export.readthedocs.io/en/latest/>

<https://www.django-rest-framework.org/>

<https://django.fun/tutorials/put-ot-request-do-response-v-django/>

<https://pythonist.ru/kastomizacziya-admin-paneli-django/>