

**Министерство науки и высшего образования Российской Федерации**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО  
ОБРАЗОВАНИЯ**

**“САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ  
ИССЛЕДОВАТЕЛЬСКИЙ  
УНИВЕРСИТЕТ ИТМО”**

**Лабораторная работа №2-3**

Предмет: Основы Web-программирования

**РЕАЛИЗАЦИЯ WEB-СЕРВИСОВ СРЕДСТВАМИ Django REST  
framework и Vue.js**

Выполнила: Евшина Яна,  
группа К3342

Проверил: Говоров А. И.

Санкт-Петербург,  
2020 г.

**Цель:** овладеть практическими навыками и умениями реализации web-сервисов средствами Django REST framework, Vue.js, Muse-UI.

**Программное обеспечение:** Python 3.6, Django REST framework, Vue.js, Muse-UI.

**Практическое задание:**

Реализовать сайт, используя вышеуказанные технологии, в соответствии с практическим заданием.

**Этапы работы:**

1. Реализовать модель данных, согласовать с преподавателем.
2. Согласовать с преподавателем список интерфейсов для лабораторной работы.
3. Реализовать серверную и клиентскую часть.

По результатам работы необходимо подготовить отчет в форме текстового документа.

**Задание, вариант 4**

Создать программную систему, предназначенную для организаторов ежегодных выставок собак. Выставки могут быть моно- и полипородные. Она должна обеспечивать хранение сведений о собаках - участниках выставок и экспертах.

Для каждой собаки в БД должны храниться сведения, о том, к какому клубу она относится, кличка, порода и возраст, классность, дата последней прививки, фамилия, имя, отчество хозяина. Перед соревнованиями собаки должны пройти обязательный медосмотр.

Т.к. участие является платным, то хозяин обязан после регистрации должен оплатить счет и предоставить его организаторам.

Сведения об эксперте должны включать фамилию и имя, номер ринга, который он обслуживает. Каждый ринг могут обслуживать несколько экспертов.

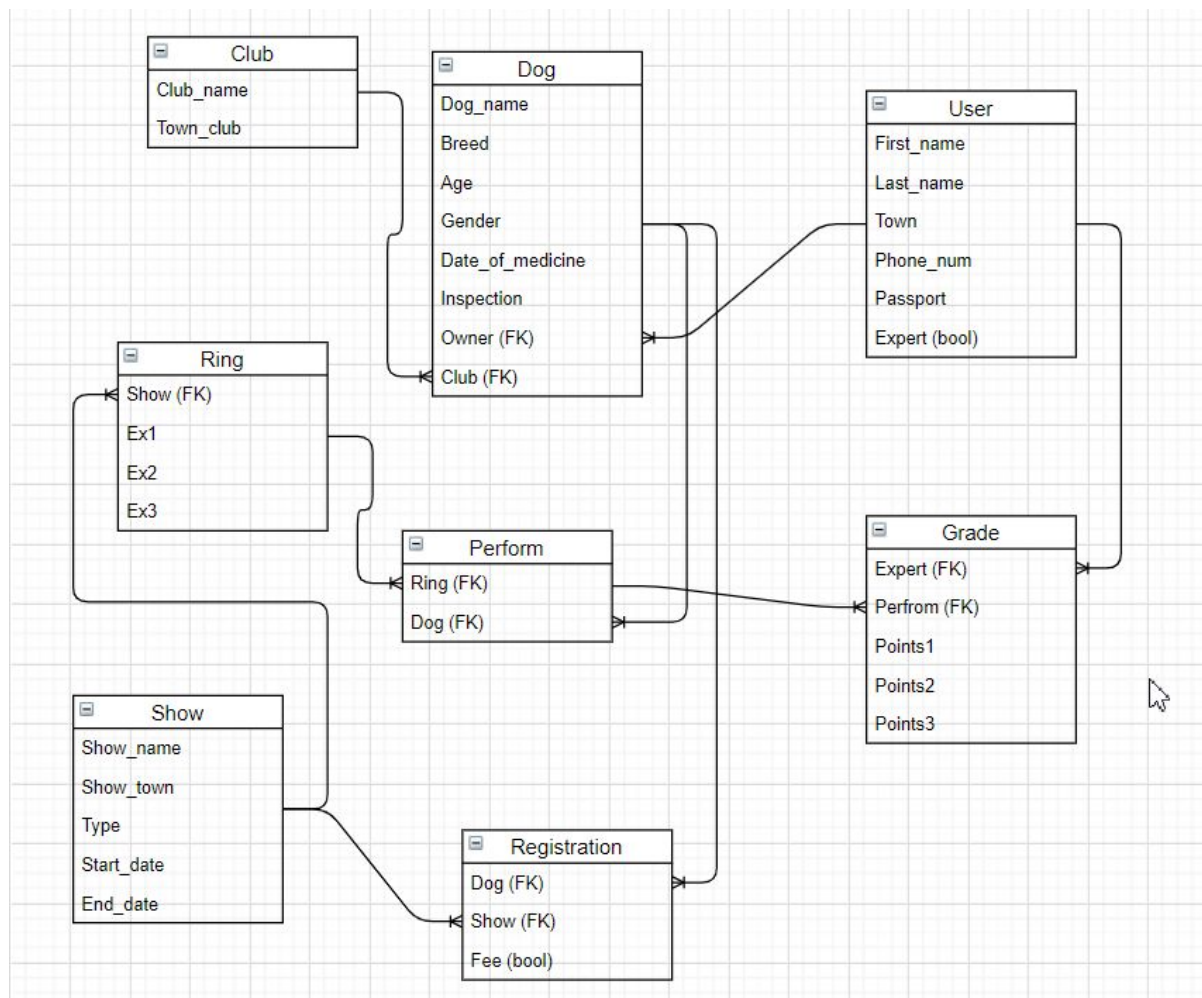
Каждая собака должна выполнить 3 упражнения, за каждое из которых она получает баллы от каждого эксперта. Итогом выставки является определение медалистов по каждой породе по итоговому рейтингу.

Организатор выставки должен иметь возможность добавить в базу нового участника или нового эксперта, снять эксперта с судейства, заменив его другим, отстранить собаку от участия в выставке.

Необходимо предусмотреть возможность выдачи отчета о результатах заданной выставки.

Ход работы.

Сначала была создана модель базы данных.



Получилось 8 таблиц:

Club	Клуб
Dog	Собака
User	Пользователь
Ring	Ринг на выставке
Show	Сама выставка
Registration	Регистрация на выставку
Grade	Оценка эксперта
Perform	Выступление собаки на ринге

Пример одной из созданных моделей, Dog:

```
class Dog(models.Model):
    dog_name = models.CharField(max_length=25)
    breed = models.CharField(max_length=25)
    age = models.IntegerField()
    gender_choice = [
        ('M', 'Male'),
        ('F', 'Female'),
    ]
    gender = models.CharField(max_length=2, choices=gender_choice)
    date_of_medicine = models.DateField()
    inspection = models.BooleanField(
        verbose_name='Medicine inspection done', default=False)
    owner = models.ForeignKey(settings.AUTH_USER_MODEL, on_delete=models.PROTECT,
                              null=True)
    club = models.ForeignKey(Club, on_delete=models.PROTECT,
                              null=True)

    def __str__(self):
        return self.dog_name
```

Также для бэкэнда был использован Django Rest Framework для более удобной работы. Были реализованы сериалайзеры, например, для вывода списков собак и клубов

```
class DogListSerializer(serializers.ModelSerializer):
    class Meta:
        model = Dog
        fields = "__all__"

class ClubListSerializer(serializers.ModelSerializer):
    class Meta:
        model = Club
        fields = "__all__"
```

Запустив DRF сервер, можно увидеть, что они работают в соответствии с заданными url.

```
{
  "id": 1,
  "owner": "Admin",
  "club": "Beautiful",
  "dog_name": "Lucky",
  "breed": "labrador",
  "age": 3,
  "gender": "M",
  "date_of_medicine": "2020-05-27",
  "inspection": false
}
```

Dog name	<input type="text" value="Peach"/>
Breed	<input type="text" value="pomeranian"/>
Age	<input type="text" value="4"/>
Gender	<input type="text" value="Female"/>
Date of medicine	<input type="text" value="20.04.2020"/>
Medicine inspection done	<input checked="" type="checkbox"/>

Можно также добавить свою собаку, введя данные в html форму внизу страницы. В файле views.py вывод выглядит так:

```
class DogDetailView(generics.RetrieveAPIView):
    queryset = Dog.objects.all()
    serializer_class = DogListSerializer

class ClubListView(generics.ListAPIView, generics.CreateAPIView):
    queryset = Club.objects.all()
    serializer_class = ClubListSerializer
```

Файл urls.py:

```
urlpatterns = [
    path('dogs/', views.DogListView.as_view()),
    path('dogs/<int:pk>/', views.DogDetailView.as_view()),

    path('users/register/', views.RegistrationAPI.as_view(), name='register'),
    path('users/profile-info/', views.UserDetailedView.as_view(), name='profile-info'),

    path('clubs/', views.ClubListView.as_view()),
    path('experts/', views.ExpertListView.as_view()),

    path('shows/', views.ShowListView.as_view()),
    path('shows/<int:pk>/rings/', views.RingListView.as_view(), name='rings'),
    path('shows/<int:pk>/rings/<int:ring_id>/grades/', views.GradeListView.as_view()),

    path('registrations/', views.RegistrationListView.as_view()),

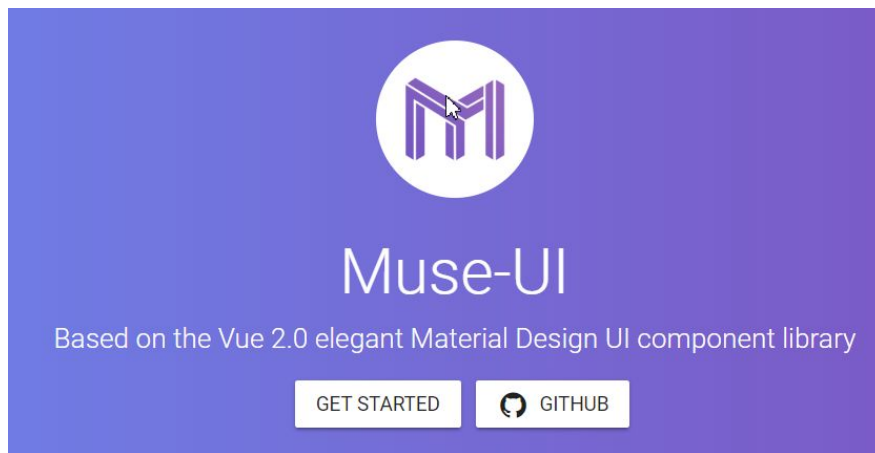
    path('performs/', views.PerformListView.as_view()),
]
```

По названиям url в целом понятно, что они выводят.

Вот так выглядит наша Django админка:

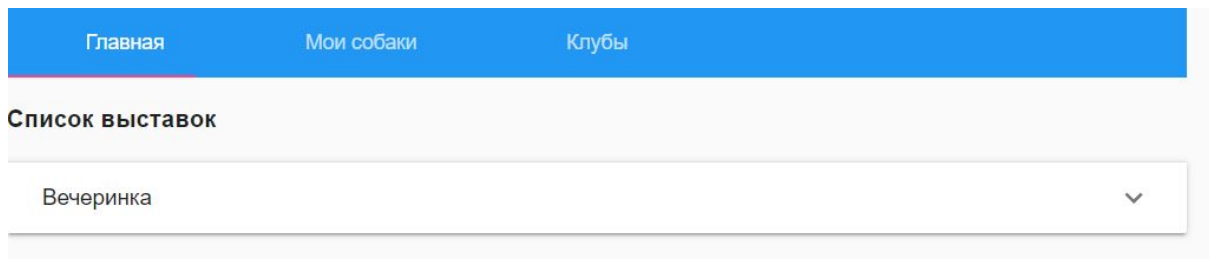
DOGS		
Clubs	+ Add	Change
Dogs	+ Add	Change
Grades	+ Add	Change
Performs	+ Add	Change
Registrations	+ Add	Change
Rings	+ Add	Change
Shows	+ Add	Change
Users	+ Add	Change

Для фронтэнда был использован vue.js и также muse.ui, для упрощения работы.

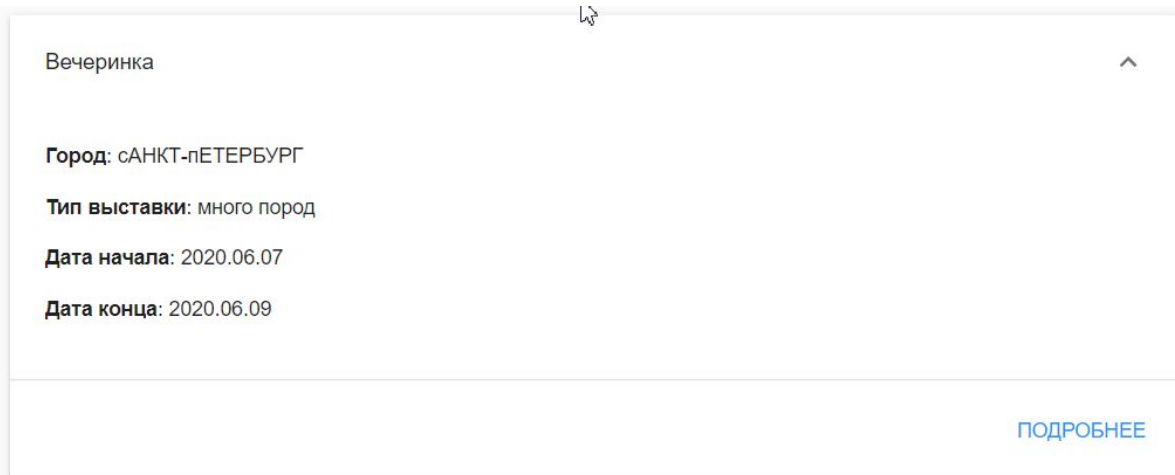


По итогу, получилось у нас очень даже ничего.

На главной страничке нашего сайта с выставками у нас выводится список имеющихся выставок.



Можно посмотреть подробнее информацию о каждой выставке, в формате раскрывающегося списка.



Можно тыкнуть на подробнее, чтобы подробнее ознакомиться с выставкой и зарегистрироваться на нее.

Подать заявку на участие в шоу

Собака

ПОДАТЬ ЗАЯВКУ

Вот так выглядит форма регистрации на выставку. Она доступна только авторизованным пользователям, которые добавили в свой личный кабинет свою собаку.

Простенькие авторизация и регистрация:

## Авторизация

username

---

Username must be filled in

password

---

## Регистрация

username

---

help text

last name

---

help text

password

---

password 2

---



Вот через такую форму можно добавить свою собаку:

The form is titled "Добавить собаку" (Add dog) and is part of a web application with a blue header containing navigation links: "Главная" (Home), "Мои собаки" (My dogs), and "Клубы" (Clubs). The form fields include: "Кличка" (Name) and "Порода" (Breed) as text inputs; "Возраст" (Age) as a range slider; "Дата прививки" (Vaccination date) as a date input; "Пол" (Sex) with radio buttons for "М" (Male) and "Ж" (Female); "Клуб" (Club) as a dropdown menu; and "Мед. осмотр" (Medical check) with radio buttons for "Да" (Yes) and "Нет" (No). A blue "ДОБАВИТЬ" (Add) button is located below the form fields. Below the form is a section titled "Список моих собак" (List of my dogs) which currently contains one entry, "Лорд" (Lord).

После добавления можем выбрать собаку для регистрации на выставку

**Подать заявку на участие в шоу**

Собака

Лорд

ПОДАТЬ ЗАЯВКУ

После того, как администратор одобрит нам заявку на выставку и распределит на ринг, мы сможем увидеть подробную информацию о выставке:

#### Информация о шоу

Город: САНКТ-ПЕТЕРБУРГ

Тип выставки: много пород

Дата начала: 2020.06.07

Дата конца: 2020.06.09

#### Ринги

##### Ринг 1

Упражнение 1: Стойка

Упражнение 2: Показать зубы

Упражнение 3: Дать лапу

#### Результаты

Кличка	1 упр	2 упр	3 упр	Итого
Лорд	8	9	10	9

Тут доступна информация, список рингов. Также задания для каждого ринга и оценка экспертами каждой собаки. Оценка считается средняя арифметическая и сортируется в порядке убывания.

Также можно изучить список клубов:

Главная

Мои собаки

Клубы

Список клубов

Beautiful

Город: Moscow

August

#### Вывод:

Был реализован сайт с использованием Django, DRF и Vue.js в соответствии с практическим заданием.