

**Санкт-Петербургский национальный исследовательский университет  
информационных технологий, механики и оптики**

**Отчет по практической работе №1  
по дисциплине «Основы веб-программирования»**

**Подготовила: Жабровец Екатерина**

**Группа: К3343**

**Преподаватель: Говоров Антон Игоревич**

**2020**

Цель: ознакомление с архитектурой и возможностями Django Web Framework

Задачи:

- 1) Создание модели Django
- 2) Создание админ панели для разработанной модели данных
- 3) Создание контроллеров для обработки данных
- 4) Настройка адресации

Задача 1 - Создание модели Django

Содержимое файла models.py

```
from django.db import models

# Create your models here.


class Owner(models.Model):
    first_name = models.CharField(max_length=50)
    last_name = models.CharField(max_length=50)
    birth_date = models.DateField()

    def __str__(self):
        return self.first_name

    def __str__(self):
        return self.last_name


class License(models.Model):
    TYPE_LICENSE = (
        ('A1', 'A'),
        ('B1', 'B'),
        ('C1', 'C'),
        ('D1', 'D'),
        ('E1', 'E'),
        ('T1', 'Tm'),
        ('T2', 'Tb'),
    )
    license_number = models.CharField(max_length=6)
    begin_date = models.DateField()
    type = models.CharField(max_length=3, choices=TYPE_LICENSE)
    owner = models.ForeignKey(Owner, on_delete=models.CASCADE)
```

```

def __str__(self):
    return self.license_number

def __str__(self):
    return self.type

class Car(models.Model):
    brand = models.CharField(max_length=50)
    model = models.CharField(max_length=50)
    color = models.CharField(max_length=50)
    number = models.CharField(max_length=6)
    own = models.ManyToManyField(Owner, through='Ownership')

    def __str__(self):
        return self.brand

    def __str__(self):
        return self.model

    def __str__(self):
        return self.color

    def __str__(self):
        return self.number

class Ownership(models.Model):
    owner = models.ForeignKey(Owner, on_delete=models.CASCADE)
    car = models.ForeignKey(Car, on_delete=models.CASCADE)
    start_date = models.DateField()
    end_date = models.DateField()

```

## Задача 2 - Создание админ панели для разработанной модели данных

Отображение таблиц в админ-панели:

Django administration
WELCOME, LUMISADE VIEW SITE / CHANGE PASSWORD / LOG OUT

Site administration

AUTHENTICATION AND AUTHORIZATION

Groups	+ Add	Change
Users	+ Add	Change

PROJECT\_FIRST\_APP

Cars	+ Add	Change
Licenses	+ Add	Change
Owners	+ Add	Change
Ownships	+ Add	Change

Recent actions

My actions

- + Zhabrovets  
Owner
- + Ownership object (2)  
Ownership
- + 766787  
Car
- + T1  
License

### Задача 3 - Создание контроллеров для обработки данных

#### Содержимое файла views.py

```
from django.http import Http404, HttpResponse
from django.shortcuts import render
from project_first_app.models import Owner

def detail(request, owner_id):
    try:
        o = Owner.objects.get(pk=owner_id)
    except Owner.DoesNotExist:
        raise Http404("Owner does not exist")
    return render(request, "Owner.html", {'owner': o})
```

#### Содержимое файла Owner.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Owners</title>
</head>
<body>
    Имя: {{ owner.first_name }}, Фамилия: {{ owner.last_name }}
</body>
</html>
```

### Задача 4 – Настройка адресации

#### Содержимое файла app\_urls.py (файл юрлов в папке приложения)

```
from django.urls import path
from . import views

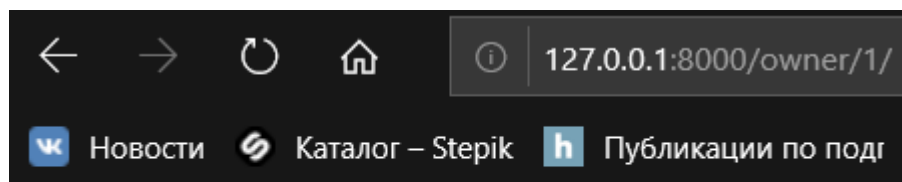
urlpatterns = [
    path('owner/<int:owner_id>/', views.detail, name='detail'),
    path('blog/', views.post_list, name='post_list')
]
```

#### Содержимое файла urls.py (файл юрлов в папке проекта)

```
from django.contrib import admin
from django.urls import path, include

urlpatterns = [
    path('admin/', admin.site.urls),
    path('', include('project_first_app.app_urls'), name='include'),
]
```

Результат



Имя: Ekaterina, Фамилия: Zhabrovets