САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ

ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ

ОТЧЕТ

ПО ПРАКТИЧЕСКОЙ РАБОТЕ №1

ПО ДИСЦИПЛИНЕ «ОСНОВЫ WEB-РАЗРАБОТКИ»

Выполнила:

Сальникова Надежда

группа К3342

Преподаватель:

Говоров А.И.

2019

1. **Цель работы:**

Овладеть базовыми навыками работы с Django – фреймворком python.

2. **Ход работы:**

Задача 1. Создание модели Django

Была создана модель Django, состоящая из 4х таблиц – автовладелец, машина, водительское удостоверение и владение машиной. Таблицы связаны между собой с использованием внешних сущностей (ForeignKey). Код из файла models.py можно увидеть на скриншотах, приведенных ниже.

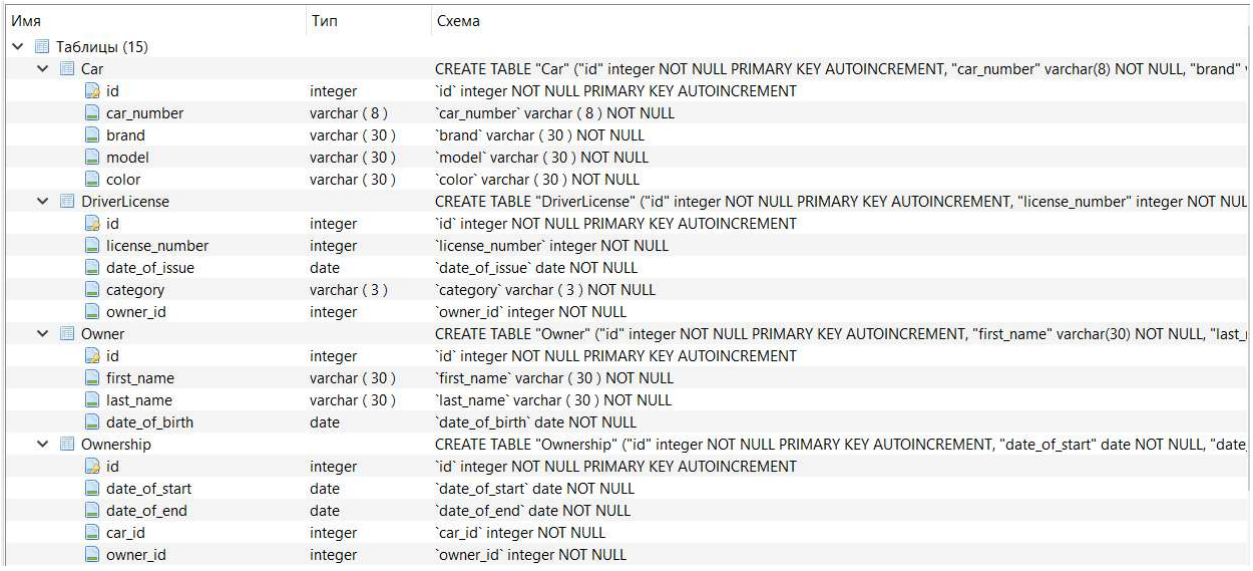
```python
from django.db import models


class Owner(models.Model):

    class Meta:
        db_table = 'Owner'

    first_name = models.CharField(max_length=30)
    last_name = models.CharField(max_length=30)
    date_of_birth = models.DateField()

    def __str__(self):
        return self.last_name


class Car(models.Model):

    class Meta:
        db_table = 'Car'

    car_number = models.CharField(max_length=8)
    brand = models.CharField(max_length=30)
    model = models.CharField(max_length=30)
    color = models.CharField(max_length=30)

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

```python
class DriverLicense(models.Model):

    class Meta:
        db_table = 'DriverLicense'

    CLASSES_OF_LICENSES = (...)
    license_number = models.IntegerField()
    date_of_issue = models.DateField()
    owner = models.ForeignKey(Owner, on_delete=models.CASCADE)
    category = models.CharField(max_length=3, choices=CLASSES_OF_LICENSES)


class Ownership(models.Model):

    class Meta:
        db_table = 'Ownership'

    owner = models.ForeignKey(Owner, on_delete=models.CASCADE)
    car = models.ForeignKey(Car, on_delete=models.CASCADE)
    date_of_start = models.DateField()
    date_of_end = models.DateField()
```

В INSTALLED_APPS было указано созданное приложение.

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

Далее с помощью команд makemigartion и migrate изменения зафиксированные в файле models были применены к базе данных. Для ее просмотра использовалось ПО DB browser for SQLite, скриншот с созданными таблицами из которого приведен далее.



Имя	Тип	Схема
Таблицы (15)		
Car		CREATE TABLE "Car" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "car_number" varchar(8) NOT NULL, "brand"
id	integer	`id` integer NOT NULL PRIMARY KEY AUTOINCREMENT
car_number	varchar (8)	`car_number` varchar (8) NOT NULL
brand	varchar (30)	`brand` varchar (30) NOT NULL
model	varchar (30)	`model` varchar (30) NOT NULL
color	varchar (30)	`color` varchar (30) NOT NULL
DriverLicense		CREATE TABLE "DriverLicense" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "license_number" integer NOT NUL
id	integer	`id` integer NOT NULL PRIMARY KEY AUTOINCREMENT
license_number	integer	`license_number` integer NOT NULL
date_of_issue	date	`date_of_issue` date NOT NULL
category	varchar (3)	`category` varchar (3) NOT NULL
owner_id	integer	`owner_id` integer NOT NULL
Owner		CREATE TABLE "Owner" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "first_name" varchar(30) NOT NULL, "last_
id	integer	`id` integer NOT NULL PRIMARY KEY AUTOINCREMENT
first_name	varchar (30)	`first_name` varchar (30) NOT NULL
last_name	varchar (30)	`last_name` varchar (30) NOT NULL
date_of_birth	date	`date_of_birth` date NOT NULL
Ownership		CREATE TABLE "Ownership" ("id" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "date_of_start" date NOT NULL, "date
id	integer	`id` integer NOT NULL PRIMARY KEY AUTOINCREMENT
date_of_start	date	`date_of_start` date NOT NULL
date_of_end	date	`date_of_end` date NOT NULL
car_id	integer	`car_id` integer NOT NULL
owner_id	integer	`owner_id` integer NOT NULL

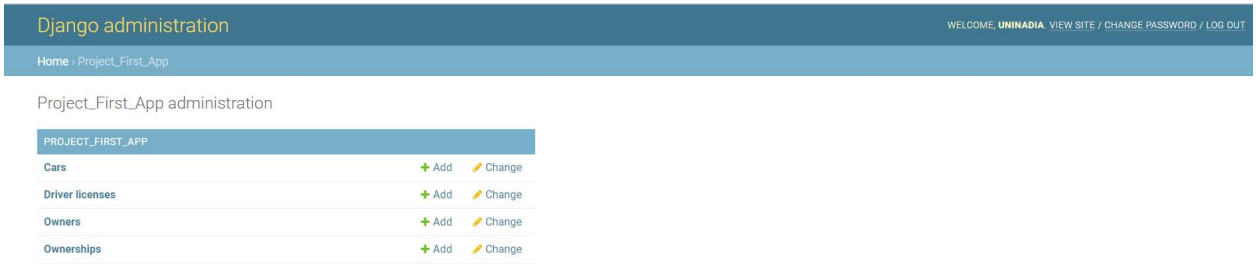
Задача 2.

Далее был модифицирован файл admin.py, который отвечает за создание и работу панели администратора. В данный файл были добавлены средства для отображения и регистрации данных. Код файла admin.py можно увидеть на следующем скриншоте.

```python
from django.contrib import admin
from project_first_app.models import Owner, Car, DriverLicense, Ownership


class OwnerAdmin(admin.ModelAdmin):
    list_display = ('first_name', 'last_name', 'date_of_birth')
    list_display_links = ('first_name', 'last_name', 'date_of_birth')


class CarAdmin(admin.ModelAdmin):
    list_display = ('car_number', 'brand', 'model', 'color')
    list_display_links = ('car_number', 'brand', 'model', 'color')


class DriverLicenseAdmin(admin.ModelAdmin):
    list_display = ('license_number', 'date_of_issue', 'category', 'owner')
    list_display_links = ('license_number', 'date_of_issue', 'category', 'owner')


class OwnershipAdmin(admin.ModelAdmin):
    list_display = ('owner', 'car', 'date_of_start', 'date_of_end')
    list_display_links = ('owner', 'car', 'date_of_start', 'date_of_end')


admin.site.register(Owner, OwnerAdmin)
admin.site.register(Car, CarAdmin)
admin.site.register(DriverLicense, DriverLicenseAdmin)
admin.site.register(Ownership, OwnershipAdmin)
```

Далее был создан суперпользователь с помощью команды createsuperuser и запущен сам сервер через runserver.

На локальном сервере был получен следующий результат.



После запуска сервера с использование панели администратора в систему были добавлены 2 владельца автомобилей, 4 автомобиля, а так же для каждого владельца было создано по 3 связи с автомобилем через сущность владение.

Далее приведены скриншоты созданных данных.

Созданные автомобили.

CAR NUMBER	BRAND	MODEL	COLOR
K931MT	Toyota	Camry	silver
B235BT	Volkswagen	Golf	red
A754KL	Toyota	Corolla	white
C065MK	Honda	Civic	black

4 cars

Созданные автовладельцы.

FIRST NAME	LAST NAME	DATE OF BIRTH
Petr	Petrov	Aug. 11, 1989
Ivan	Ivanov	March 23, 1994

2 owners

Созданные владения.

OWNER	CAR	DATE OF START	DATE OF END
Petrov	A754KL	Dec. 4, 2018	Jan. 27, 2020
Petrov	K931MT	Nov. 25, 2013	May 28, 2018
Petrov	B235BT	March 27, 2007	Oct. 1, 2013
Ivanov	K931MT	May 30, 2011	Nov. 24, 2013
Ivanov	A754KL	March 16, 2015	Sept. 8, 2018
Ivanov	C065MK	Feb. 10, 2014	Feb. 14, 2015

6 ownerships

Задача 3.

В файле views.py был создан контроллер owner_detail, который позволяет по id номеру владельца отобразить информацию о нем с использование html шаблона.

```python
from django.http import Http404
from django.shortcuts import render
from project_first_app.models import Owner, Car, DriverLicense, Ownership


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

Также был создан шаблон owner.html, размещенный в папке templates.

```
<html>
<body>
    Firts name: {{owner.first_name}}</br>
    Last name: {{owner.last_name}}</br>
    Birth date: {{owner.date_of_birth}}
</body>
```

Для обращения к этой директории в папку с настройками были доавблены следующие строки.

После описания основной директории проекта

```
TEMPLATE_DIR = os.path.join(BASE_DIR, 'templates')
```

И эта директория была указана в пункте DIRS.

```
TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [TEMPLATE_DIR],
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [
                'django.template.context_processors.debug',
                'django.template.context_processors.request',
                'django.contrib.auth.context_processors.auth',
                'django.contrib.messages.context_processors.messages',
            ],
        },
    },
]
```

Задача 4.

Были созданы файлы urls.py в корневой папке проекта и в папке приложения.

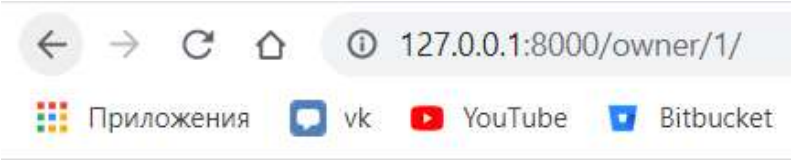
Текст urls.py из корневой папки проекта приведен ниже. Он позволяет обрабатывать url адреса приложения.

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


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

Текст urls.py из приложения приведен на следующем скриншоте. В нем вызывается контроллер owner_detail, выводящий информацию об автовладельце с соттвествующим id номером.

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


urlpatterns = [
    path('owner/<int:owner_id>/', views.owner_detail)
]
```

В результате по адресу локального сервера с добавлением owner/1/ можно увидеть информацию о первом автовладельце, как это показано ниже.



Firts name: Ivan
Last name: Ivanov
Birth date: March 23, 1994

3. Вывод:

В ходе выполнения практической работы были получены базовые навыки работы с фреймворком Django, а именно создание и применение моделей, использование панели администратора, простых контроллеров и шаблонов для представлений.