Московский государственный технический университет им. Н.Э. Баумана

Факультет «Радиотехнический»
Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Разработка интернет-приложений»

Отчет по лабораторной работе №6

Выполнил:
студент группы РТ5-51Б
Сысоев Александр

Проверил:
доцент каф. ИУ5
Гапанюк Ю. Е.

Москва, 2021 г.

ПОЛУЧЕННОЕ ЗАДАНИЕ

Цель лабораторной работы: изучение возможностей разработки REST API с использованием Django REST Framework.

Задание:

В этой лабораторной работе Вы познакомитесь с популярной СУБД MySQL, создадите свою базу данных. Также Вам нужно будет дополнить свои классы предметной области, связав их с созданной БД. После этого Вы создадите свои модели с помощью Django ORM, отобразите объекты из БД с помощью этих моделей.

1. Создайте сценарий с подключением к БД и несколькими запросами, примеры рассмотрены в методических указаниях.
2. Реализуйте модели Вашей предметной области из предыдущей ЛР (минимум две модели, т.е. две таблицы).
3. Создайте представления и шаблоны Django для отображения списка данных по каждой из сущностей.

ИСХОДНЫЙ КОД

Файл urls.py

```
from django.contrib import admin
from django.urls import include, path
from notebooks.views import NoteBookViewSet
from rest_framework import routers

router = routers.DefaultRouter()
router.register(r'notebooks', NoteBookViewSet)

# Wire up our API using automatic URL routing.
# Additionally, we include login URLs for the browsable API.
urlpatterns = [
    path('', include(router.urls)),
    path('api-auth/', include('rest_framework.urls',
namespace='rest_framework')),
    path('admin/', admin.site.urls),
]
```

Файл settings.py

```
from pathlib import Path
import os

# Build paths inside the project like this: BASE_DIR / 'subdir'.
BASE_DIR = Path(__file__).resolve().parent.parent


# Quick-start development settings - unsuitable for production
# See https://docs.djangoproject.com/en/4.0/howto/deployment/checklist/

# SECURITY WARNING: keep the secret key used in production secret!
SECRET_KEY = 'django-insecure-2u$m2)7ui@4*(um_nfj_@-hhc^dene6#bc3ut9t=73t+=-
_jbq'

# SECURITY WARNING: don't run with debug turned on in production!
DEBUG = True

ALLOWED_HOSTS = []


# Application definition

INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'notebooks.apps.NotebooksConfig',
    'rest_framework',
    'corsheaders',
]

MIDDLEWARE = [
    'django.middleware.security.SecurityMiddleware',
    'django.contrib.sessions.middleware.SessionMiddleware',
    'django.middleware.common.CommonMiddleware',
    'django.middleware.csrf.CsrfViewMiddleware',
    'django.contrib.auth.middleware.AuthenticationMiddleware',
    'django.contrib.messages.middleware.MessageMiddleware',
    'django.middleware.clickjacking.XFrameOptionsMiddleware',
    'corsheaders.middleware.CorsMiddleware',
]

CORS_ALLOWED_ORIGINS = [
    'http://localhost:3000',
]

ROOT_URLCONF = 'Lab6_1.urls'

TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [BASE_DIR / 'templates']
        ,
        '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',
            ],
```

```
        },
    },
]

WSGI_APPLICATION = 'Lab6_1.wsgi.application'


# Database
# https://docs.djangoproject.com/en/4.0/ref/settings/#databases

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': BASE_DIR / 'db.sqlite3',
    }
}


# Password validation
# https://docs.djangoproject.com/en/4.0/ref/settings/#auth-password-
validators

AUTH_PASSWORD_VALIDATORS = [
    {
        'NAME':
'django.contrib.auth.password_validation.UserAttributeSimilarityValidator',
    },
    {
        'NAME':
'django.contrib.auth.password_validation.MinimumLengthValidator',
    },
    {
        'NAME':
'django.contrib.auth.password_validation.CommonPasswordValidator',
    },
    {
        'NAME':
'django.contrib.auth.password_validation.NumericPasswordValidator',
    },
]


# Internationalization
# https://docs.djangoproject.com/en/4.0/topics/i18n/

LANGUAGE_CODE = 'en-us'

TIME_ZONE = 'UTC'

USE_I18N = True

USE_TZ = True


# Static files (CSS, JavaScript, Images)
# https://docs.djangoproject.com/en/4.0/howto/static-files/

STATIC_URL = '/static/'

STATICFILES_DIRS = [
    os.path.join(BASE_DIR, "static"),
]

# Default primary key field type
```

```
# https://docs.djangoproject.com/en/4.0/ref/settings/#default-auto-field

DEFAULT_AUTO_FIELD = 'django.db.models.BigAutoField'
```

Файл serializers.py

```
from notebooks.models import NoteBook
from rest_framework import serializers


class NoteBookSerializer(serializers.ModelSerializer):
    class Meta:
        # Модель, которую мы сериализуем
        model = NoteBook
        # Поля, которые мы сериализуем
        fields = "__all__"
```

Файл models.py

```
from django.db import models

class NoteBook(models.Model):
    manufacturer = models.CharField("Производитель", max_length=50)
    name = models.CharField("Модель", max_length=50)
    display = models.IntegerField("Диагональ дисплея")
    os = models.CharField("Операционная система", max_length=50)
    image = models.ImageField("Изображение")
```

Файл views.py

```
from rest_framework import viewsets
from notebooks.serializers import NoteBookSerializer
from notebooks.models import NoteBook


class NoteBookViewSet(viewsets.ModelViewSet):
    """
    API endpoint, который позволяет просматривать и редактировать акции
компаний
    """
    # queryset всех пользователей для фильтрации по дате последнего изменения
    queryset = NoteBook.objects.all().order_by('name')
    serializer_class = NoteBookSerializer  # Сериализатор для модели
```

РЕЗУЛЬТАТ РАБОТЫ ПРОГРАММЫ