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 «Инженерное проектирование»

Тема: «Административная панель магазина продажи электроники»

**Выполнил:**

Студент группы 191-321

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Мажаев В.С.

подпись, дата

**Принял:**

Старший преподаватель

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Даньшина М.В.

подпись, дата

Москва 2021

# СОДЕРЖАНИЕ

Оглавление

[СОДЕРЖАНИЕ 2](#_Toc76835749)

[ВВЕДЕНИЕ 3](#_Toc76835750)

[Тип разработки 3](#_Toc76835751)

[Задача 3](#_Toc76835752)

[Аналоги 3](#_Toc76835753)

[ОСНОВНАЯ ЧАСТЬ 7](#_Toc76835754)

[Структура базы данных 7](#_Toc76835755)

[Use cases 8](#_Toc76835756)

[Этапы реализации проекта 8](#_Toc76835757)

[ЗАКЛЮЧЕНИЕ 12](#_Toc76835758)

[СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ 13](#_Toc76835759)

# ВВЕДЕНИЕ

## Тип разработки

Индивидуальный проект, направленный на отработку навыков Full-stack разработки. Разработка ведется поэтапно.

## Задача

Спроектировать, создать базу данных для магазина продажи электроники и настроить административную панель. Административная панель поможет управлять сайтом, регулярно его обновлять, редактировать и создавать новые товары, акции и все что нужно магазину.

## Аналоги

Начнем с М.Видео.

В магазине есть каталог, состоящий из категорий (смартфоны, телевизоры, техника для кухни и т.д.)

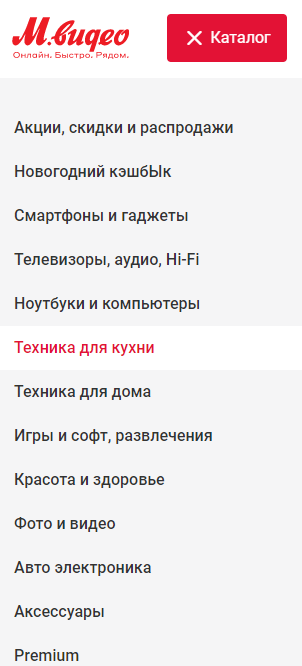


Рисунок 1

В категории находятся товары, которые обладают набором информации, а именно фотографии товара, краткая информация о товаре, его характеристики, отзывы и наличие в магазинах.

На картинке ниже раздел характеристик у смартфона.

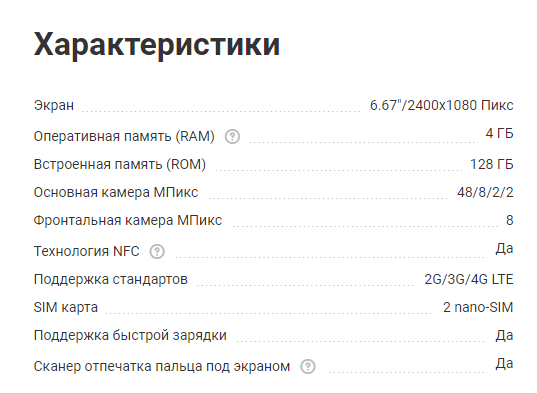


Рисунок 2

Причем количество характеристик у каждой категории товаров разное.

Отзыв состоит из имени автора, даты, количества баллов (звездочек), и текста отзыва

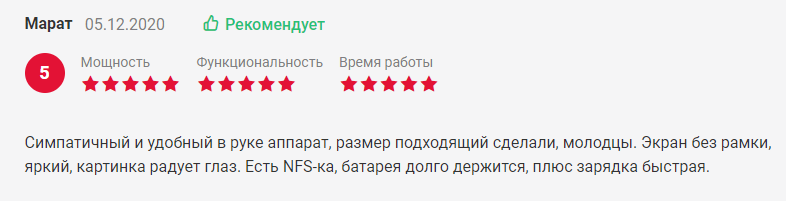


Рисунок 3

Наличие в магазинах показывается таблицей, в которой есть адрес магазина, станция метро, наличие товара и режим работы магазина

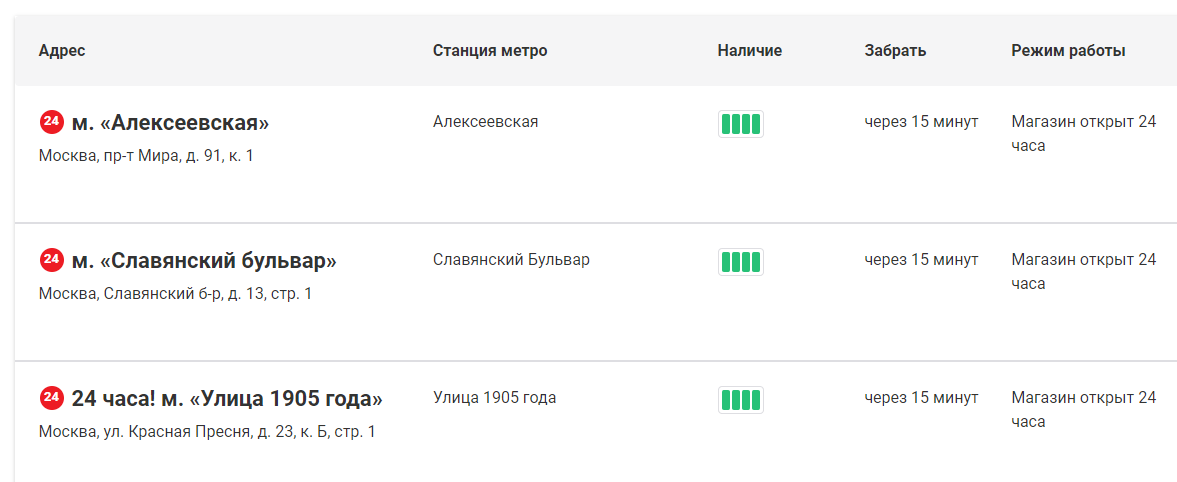


Рисунок 4

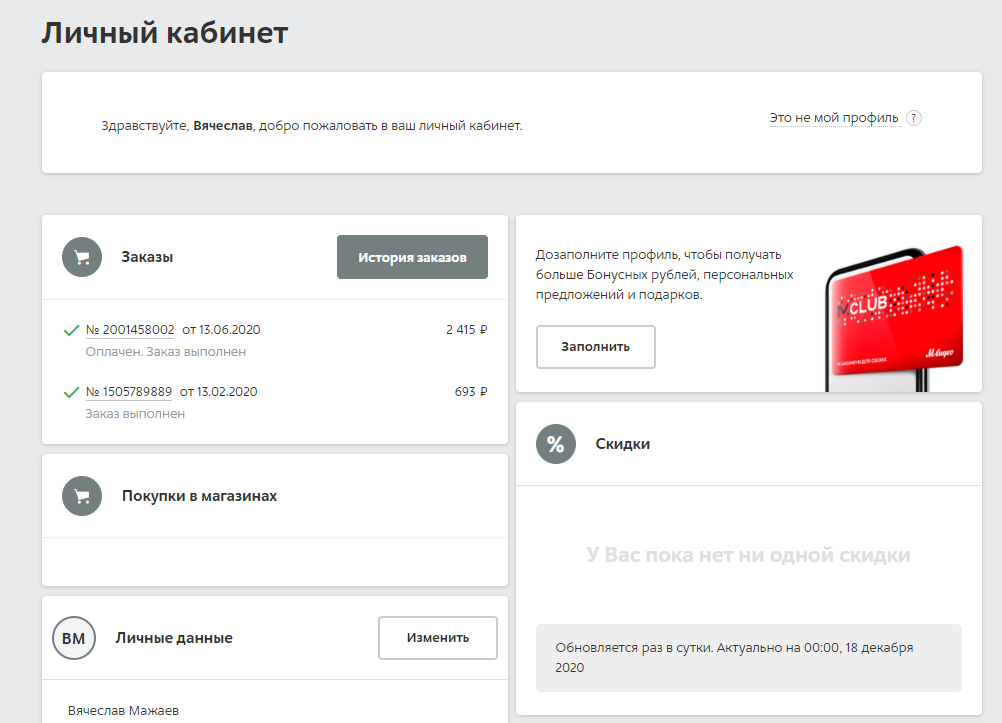
Также в этом магазине есть личный кабинет, он содержит информацию о заказах, персональных скидках и личные данные пользователя, такие как телефон, email и фио.

Рисунок 5

Также к пользователю прикреплены раздел “избранное” и корзина:

Теперь рассмотрим Ситилинк, он в общем отличается только дизайном, логические разделы те же самые.

У него тоже есть раздел с категориями

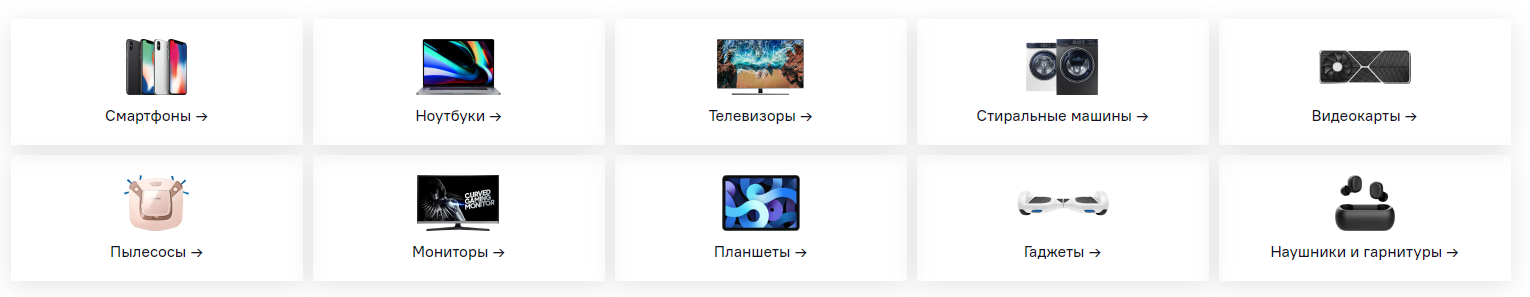


Рисунок 6

Характеристиками устройства



Рисунок 7

Отзывы к товару

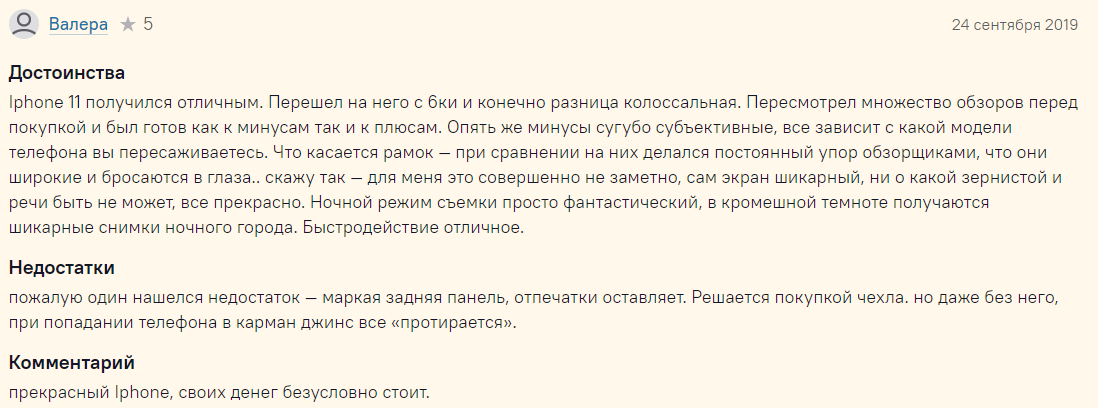


Рисунок 8

И личный кабинет имеет информацию о заказах, персональных скидках и личные данные пользователя, такие как телефон, email и ФИО. Разделы избранное и корзина также присутствуют.

# ОСНОВНАЯ ЧАСТЬ

## Структура базы данных

Следуя из анализа конкурентов, база данных и админ панель должны включать в себя сущности: товара, который имеет характеристики, относится к определенной категории товаров, имеется в наличии в магазинах в определенном количестве, имеет отзыв. Пользователь может сделать заказ, добавить товар в избранное, сделать отзыв, который включает в себя рейтинг. Рейтинг — это количество звездочек(баллов) за качество и функционал товара.

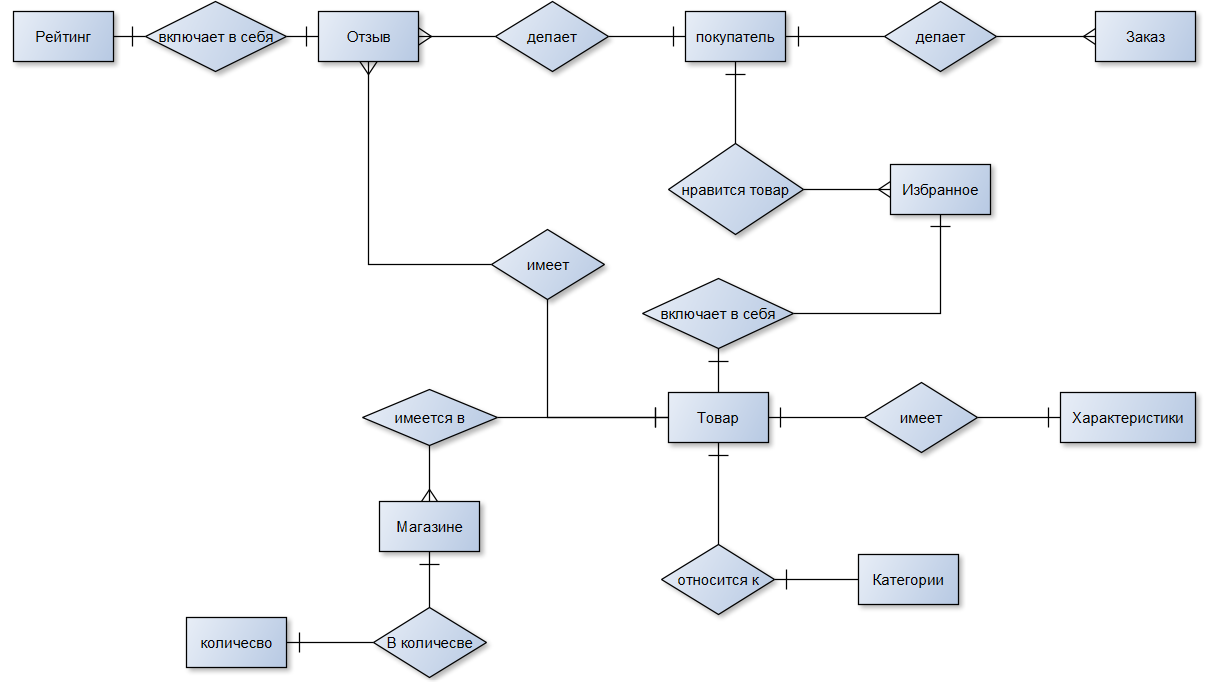


Рисунок 9 инфологическая схема БД

С помощью инфологической схемы можно составить подробную физическую схему БД. В ней находятся все те же сущности, но каждая из них имеет набор полей с определенным типом данных. Таблица products имеет связь по внешнему ключу с таблицами categories (связь один к одному) и specifications (связь один к одному). Таблица reviews имеет связь по внешнему ключу с products (связь один к одному), ratings (связь один к одному) и users (связь один ко многим). Таблицы favourite\_products и orders связаны с users и с products по внешнему ключу и имеют связь один ко многим. Таблица quantity показывает количество товара в определенном магазине и связана внешним ключом с stores (один ко многим) и products (один к одному).

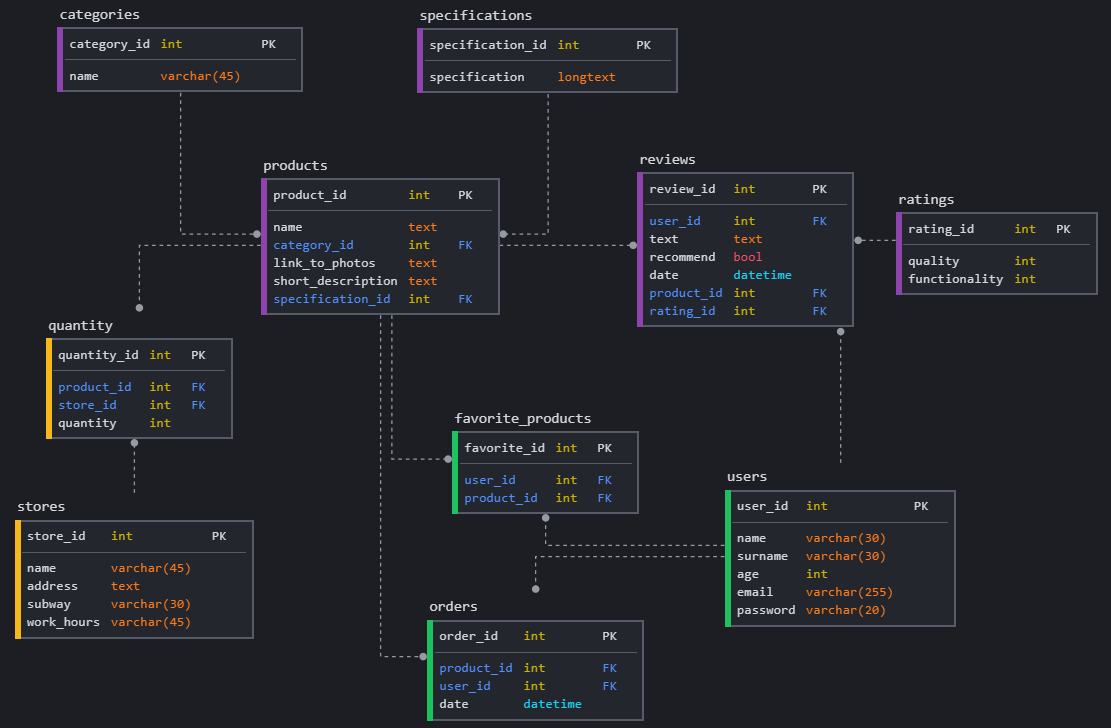


Рисунок 10 физическая схема БД

## Use cases

Кейсы использования приложения

1. Администратор добавляет новую категорию товаров
2. Администратор добавляет новый товар магазина
3. Администратор добавляет описание и характеристики к новому товару
4. Администратор добавляет новый магазин
5. Администратор импортирует данные новых товаров
6. Администратор экспортирует список отзывов от пользователей
7. Покупатель добавляет товар в избранное
8. Покупатель добавляет товар в корзину
9. Покупатель пишет отзыв к товару
10. Администратор удаляет пользователя
11. Администратор удаляет товар
12. Администратор удаляет магазин
13. Менеджер в магазине смотрит наличие товара на складе
14. Менеджер в магазине смотрит характеристики товара, чтобы рассказать покупателю
15. Разработчик проверяет работу frontend части интернет-магазина
16. Отдел логистики составляет план завоза товаров
17. Руководство компании составления отчет

## Этапы реализации проекта

1. Анализ конкурентов

Для успешной реализации проекта необходимо разбираться в области продажи электронной техники, в этом мне помог анализ конкурентов. Я проанализировал несколько сайтов занимающихся продажей электроники, а именно их структуру и функционал, чтобы понять какую информацию нужно хранить в базе данных.

1. Проектирование инфологической модели предметной области

Основываясь на накопленной информации из анализа конкурентов, я составил схему «Сущность связь», которая показывает существующие сущности в проекте и их связь.

1. Проектирование физической структуры

На основе инфологической модели я создал физическую модель базы данных, в которой видно не только сущности и их характеристики, но и тип данных.

1. Создание репозитория проекта

На GitHub я создал репозиторий проекта и склонировал его на компьютер с помощью WebStorm, далее по мере добавления функционала я создавал ветки, в которых изучал и тестировал разные фишки и вел разработку нового функционала, который мог все поломать. Только после того, как приложение могло стабильно работать я сливал ветку в мастер.

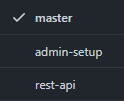


Рисунок 11 Пример веток

1. Создание Django-приложения

Для начала я изучил материалы по Django в интернете, просмотрел видео курсы и статьи, создав пару тестовых приложений. Создание основного приложения я начал с логического разделения всего проекта на приложения (products, stores, users). Они также отмечены разными цветами на физической модели базы данных. Далее я начал перенос базы данных в файлы models.py находящиеся в своих приложениях.

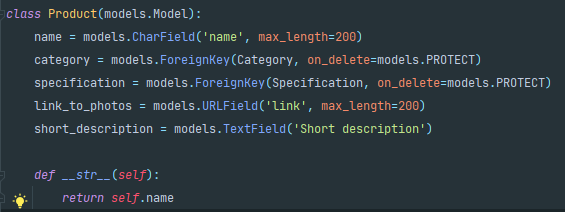


Рисунок 12 Пример сущности products

1. Настройка административного интерфейса Django

После первоначальной настройки я начал добавлять вывод сущностей в админ панель, добавил функции фильтрации и поиска

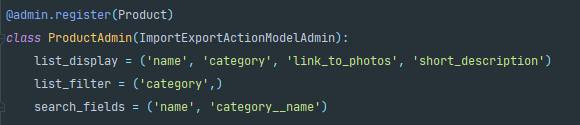


Рисунок 13 Пример регистрации сущности

Чтобы зайти в админ панель я создал два пользователя: admin/admin и user/user.

Пользователя admin я создал командой «python manage.py createsuperuser», а user создал из админ панели. Для модели user доступен только просмотр таблиц.

Вот как выглядит страница сущности products в админ панели, поиск производится по двум полям, name и category\_name причем category\_name это внешний ключ, и чтобы поиск работал с ним нужно указание таблицы, в которой находится информация, именно поэтому поле category из сущности product я вписал в search\_fields как category\_name. Фильтрация производится по категории товара.

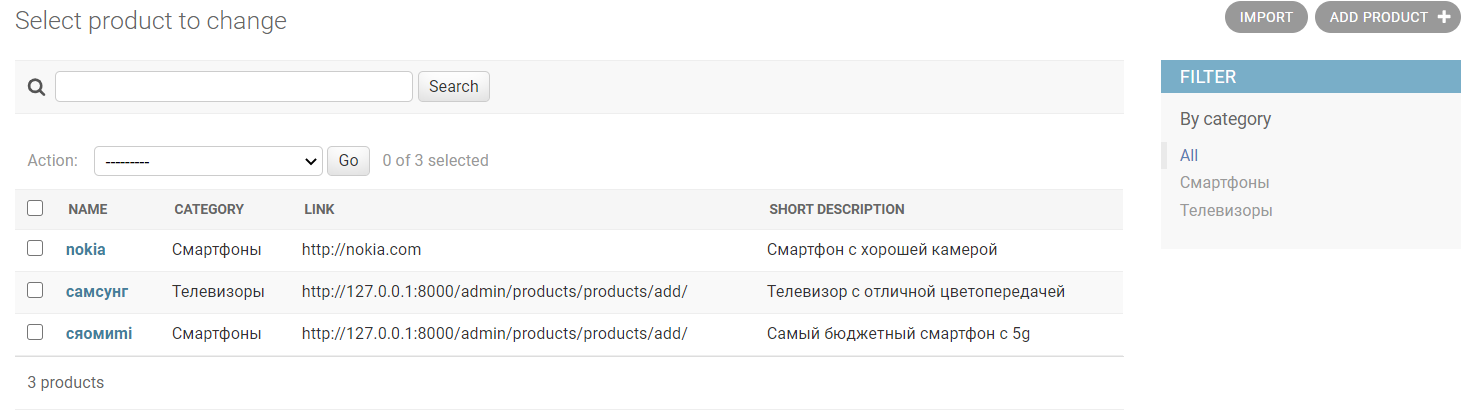


Рисунок 14

Также есть функция сортировки по возрастанию и убыванию.



Рисунок 15

Также я настроил admin actions. Я добавил функцию пополнения товаров на складе. Админ выбирает товары из списка и выполняет функцию, после этого количество выбранных товаров в магазинах становится равным 1000.

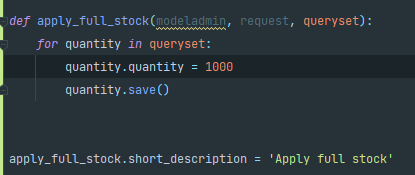


Рисунок 16

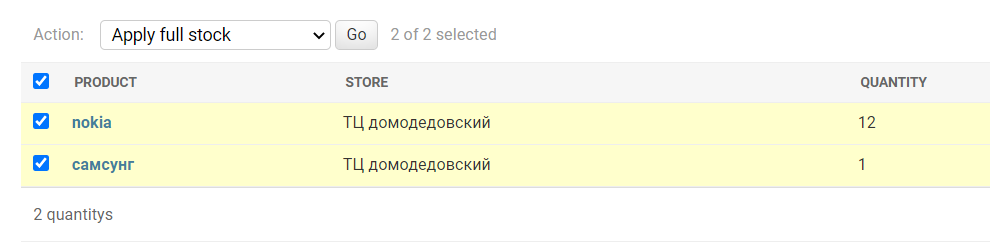


Рисунок 17

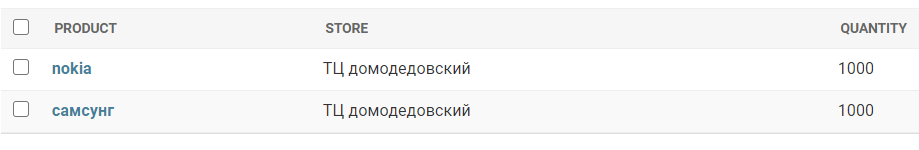


Рисунок 18

1. Наполнение базы данных

Далее я заполнил базу данных тестовой информацией

1. Реализация REST API

Я добавил возможность управления данными приложения со стороны,

реализовал REST API, а также разобрался с тестированием API через

сервис POSTMAN.

Вот 3 примера URL созданные для API

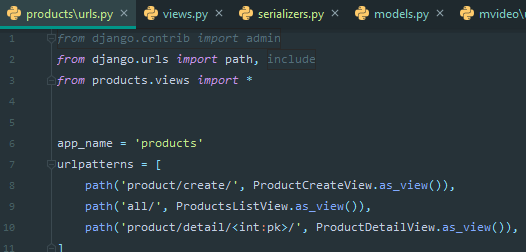


Рисунок 19

Этот URL отвечает за создание нового товара

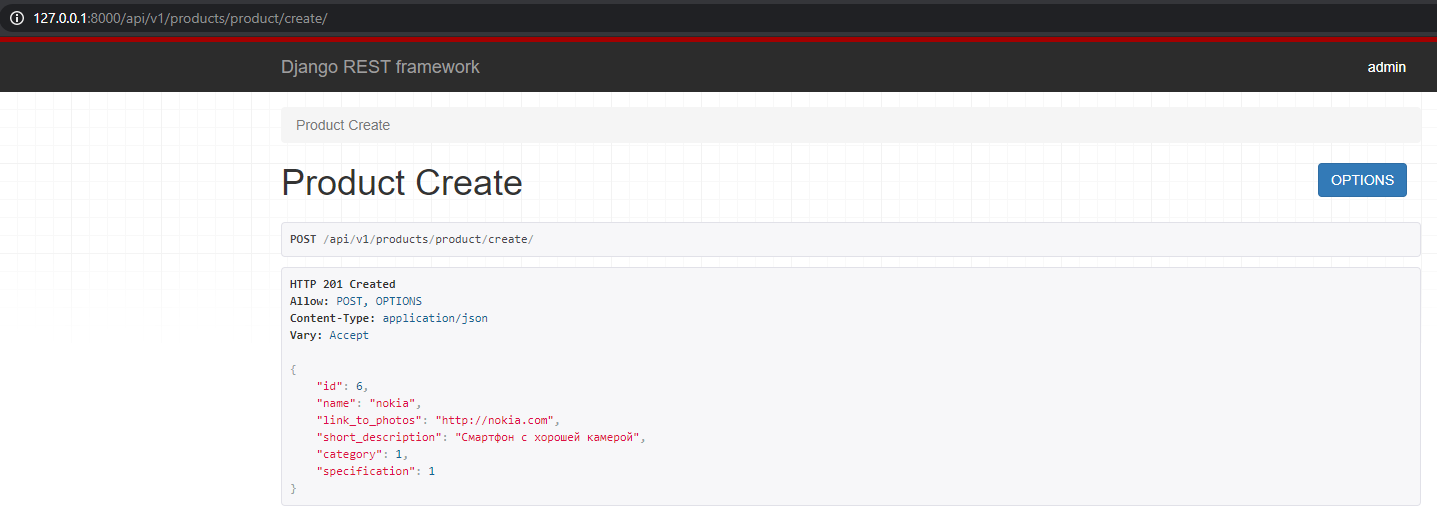


Рисунок 20

Этот URL отвечает за вывод всех товаров

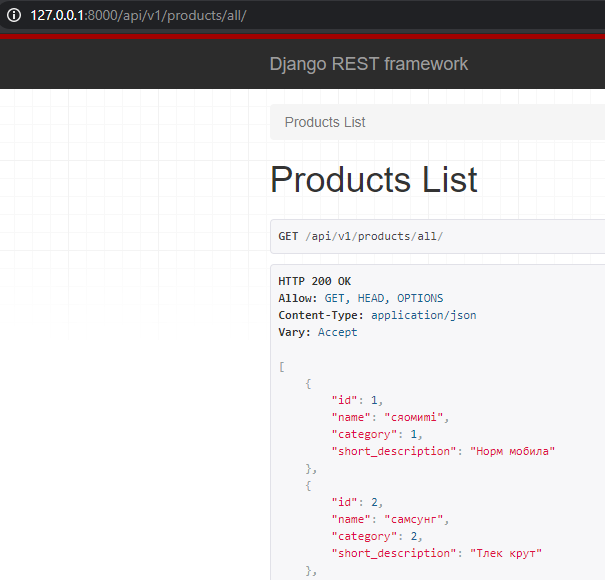


Рисунок 21

А этот за вывод товара по его id

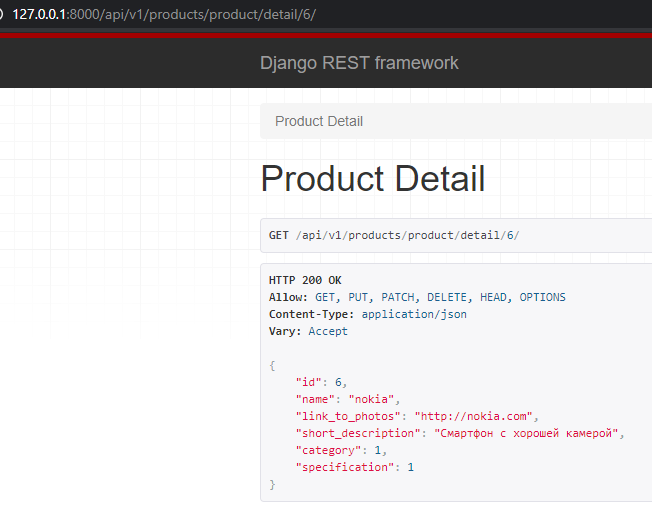


Рисунок 22

1. Документирование
2. Заполнение оценочного листа

# ЗАКЛЮЧЕНИЕ

Результатом всех работ является готовая административная панель django-приложения, два типа пользователей и мешок знаний.

GitHub <https://github.com/slavocado/mvideo> (в репозитории есть файлы отчета и оценочного листа в форматах docx и pdf)

Приложение <http://mvideo.std-947.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/>