МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКИЙ ФЕДЕРАЦИИ ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫШЕГО ОБРАЗОВАНИЯ

«МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ

(национальный исследовательской университет) (МАИ)

Кафедра «Теория вероятностей и компьютерное моделирование»

ОТЧЁТ

По программным технологиям построения управляющих оболочек систем дистанционного обучения

Тема

«Клиентское представление магазина. Представление корзины.»

Выполнил студент группы М80-101М-20:

Лозовой А.М.

Принял преподаватель кафедры 804: Жарков Е.А.

МОСКВА – 2021 год

Оглавление

[1. Постановка задачи 3](#_Toc73644460)

[2. Выбор средств разработки и их установка 4](#_Toc73644461)

[3. Структура программы 6](#_Toc73644462)

[4. Пример работы сайта 9](#_Toc73644463)

[5. Пример работы заказа 10](#_Toc73644464)

[6. Вывод 12](#_Toc73644465)

[7. Список литературы 13](#_Toc73644466)

1. Постановка задачи

**Клиентское представление магазина:**

Представление корзины:

- Список товаров внутри корзины (получение данных от сервера, можно использовать локальные данные);

- Добавление / Удаление товара (отправка POST запроса на сервер, либо mock) => Визуальное изменение страницы;

- Оформление заказа (отправка POST запроса на сервер) => Визуальное изменение страницы.

1. Выбор средств разработки и их установка

В качестве фреймворка для разработки приложения был выбран фреймворк Django, в качестве среды разработки был выбран PyCharm.

**Django** — свободный фреймворк для веб-приложений на языке Python, использующий шаблон проектирования MVC. Проект поддерживается организацией Django Software Foundation.

Сайт на Django строится из одного или нескольких приложений, которые рекомендуется делать отчуждаемыми и подключаемыми. Это одно из существенных архитектурных отличий этого фреймворка от некоторых других (например, Ruby on Rails). Один из основных принципов фреймворка — DRY (англ. Don't repeat yourself)

Также, в отличие от других фреймворков, обработчики URL в Django конфигурируются явно при помощи регулярных выражений.

Для работы с базой данных Django использует собственный ORM, в котором модель данных описывается классами Python, и по ней генерируется схема базы данных.

**PyCharm** — интегрированная среда разработки для языка программирования Python. Предоставляет средства для анализа кода, графический отладчик, инструмент для запуска юнит-тестов и поддерживает веб-разработку на Django. PyCharm разработана компанией JetBrains на основе IntelliJ IDEA.

Для установки Django, в терминале пропишем следующие команды:

*pip install Django*

*django-admin startproject* название нашего проекта

После установки Django и запуска локального сервера мы получим следующую картинку (Рис 2.1).

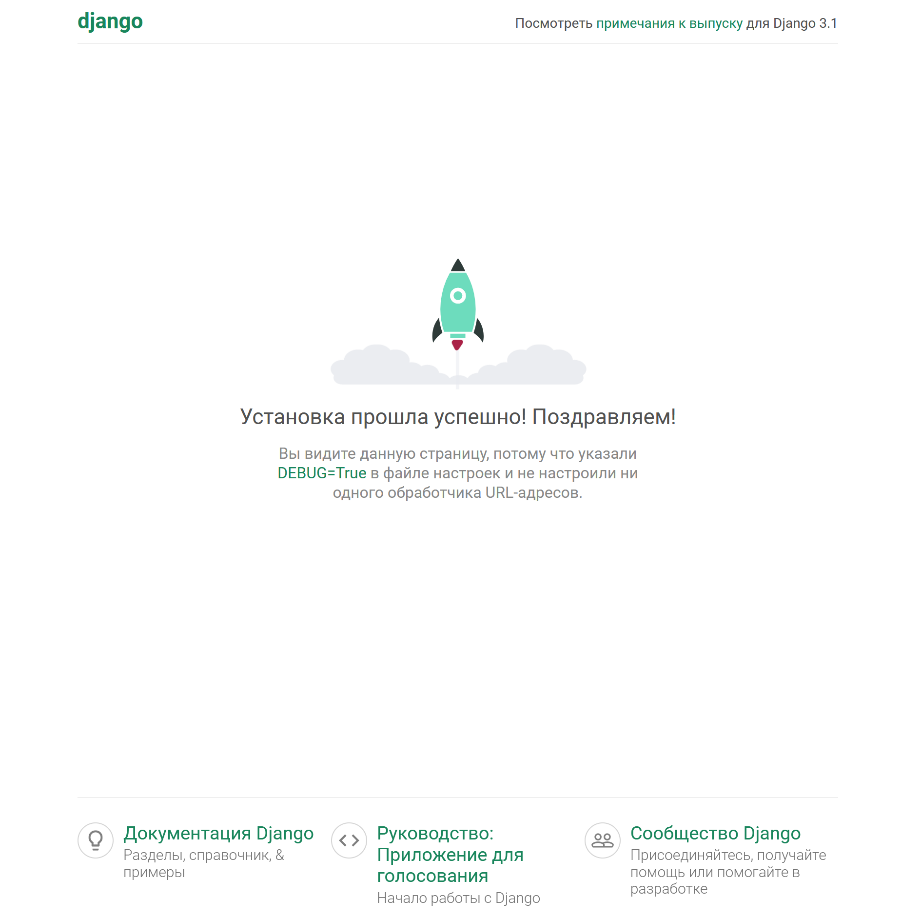


Рис. 2.1 Стартовая страница локального сервера

1. Структура программы
   1. Модели

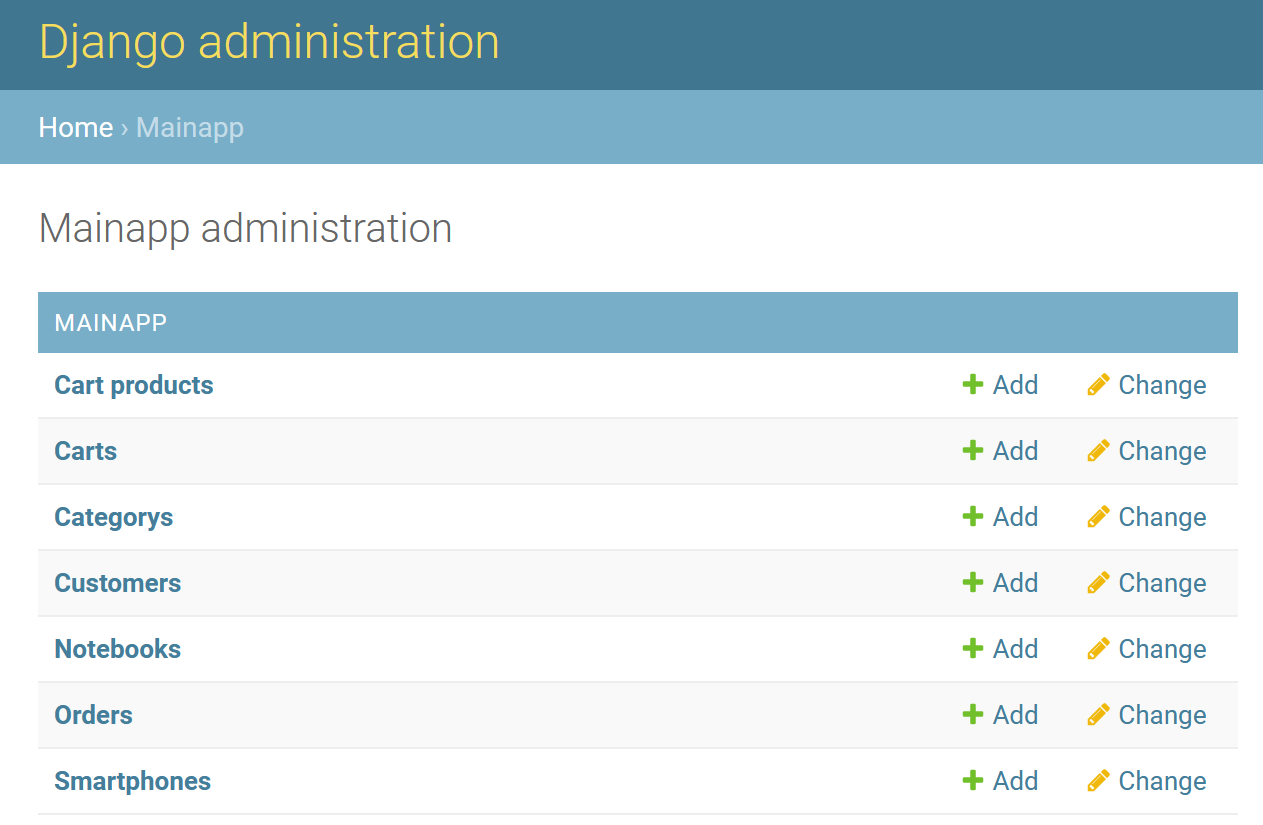


Рис. 3.1.1 Список наших моделей

**Customer** – покупатель – это модель для описания покупателя.

**Categorys** – категории – это модель для описания категорий, к которым относятся тот или иной продукт, например, ноутбук или смартфон.

**Carts** – корзины – это модель для описания корзин.

**CartProduct** – переходная модель для помещения покупок в корзину.

**Orders** – заказы – модель для описания заказов.

**Notebooks** – ноутбуки – модель для описания ноутбуков.

**Smartphones** – смартфоны – модель для описания смартфонов

* 1. Шаблон

В качестве шаблона возьмем бесплатный готовый шаблон с сайта startbootstrap.com для сайтов электронной коммерции (Рис.3.2.1).

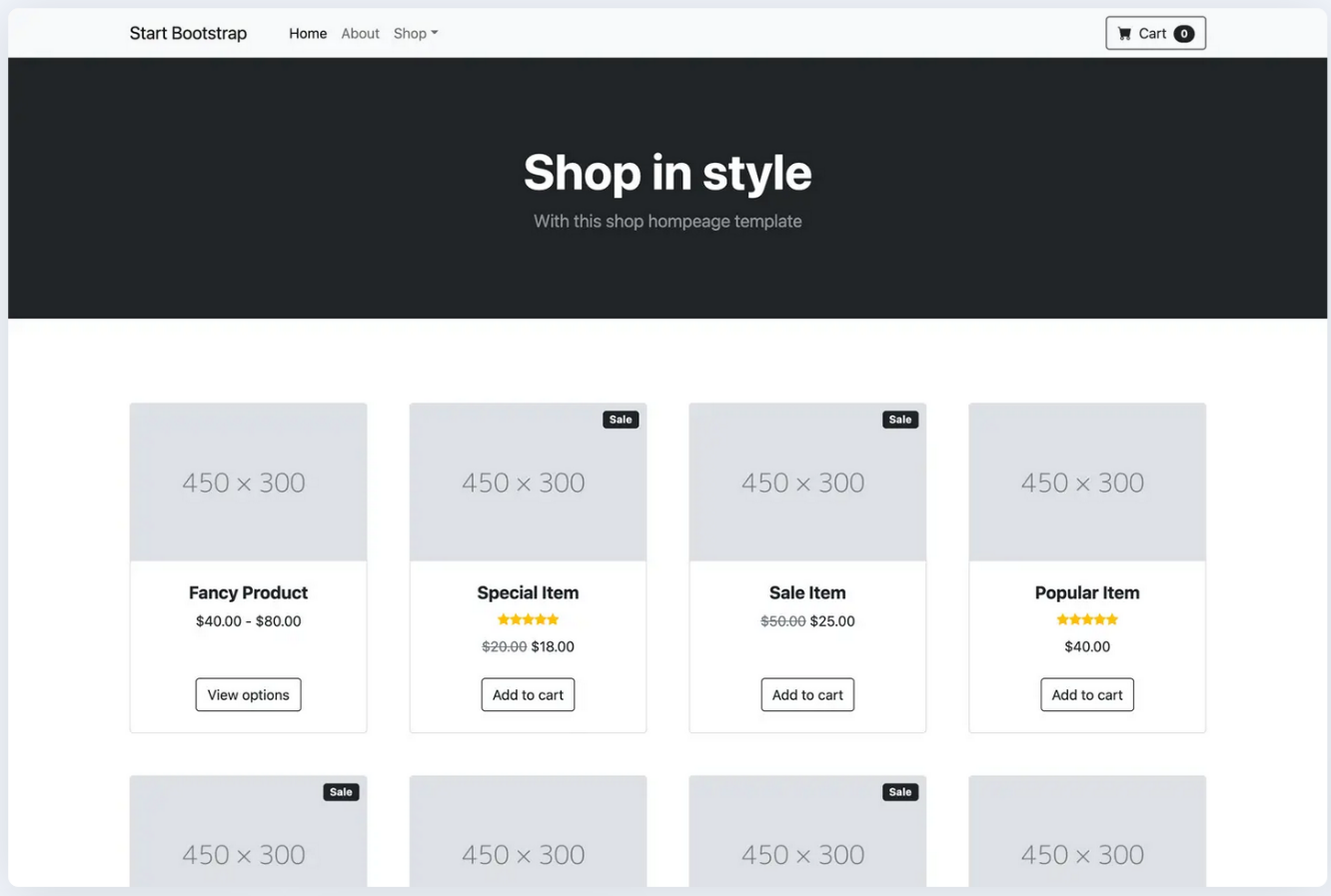


Рис. 3.2.1 Готовый шаблон сайта электронной коммерции

* 1. POST запрос

Django использует объекты запроса и ответа для передачи состояния через систему.

Когда страница запрашивается, Django создает объект HttpRequest, содержащий метаданные о запросе. Затем Django загружает соответствующее представление, передавая HttpRequest в качестве первого аргумента функции представления. Каждое представление отвечает за возврат объекта HttpResponse.

Этот документ объясняет API для объектов HttpRequest и HttpResponse, которые определены в модуле django.http.

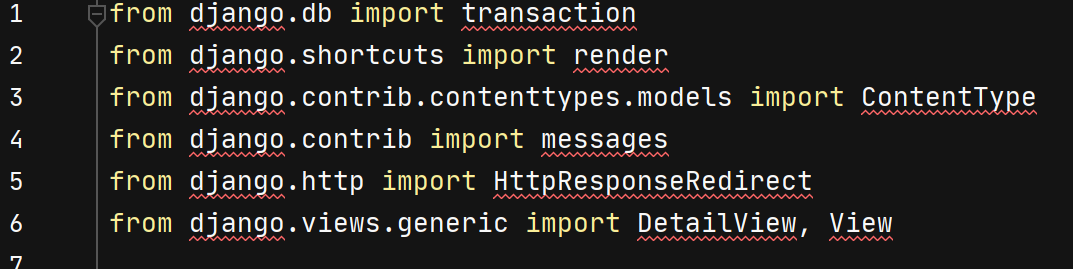


Рис.3.3.1 Пример использования Post запроса в программе

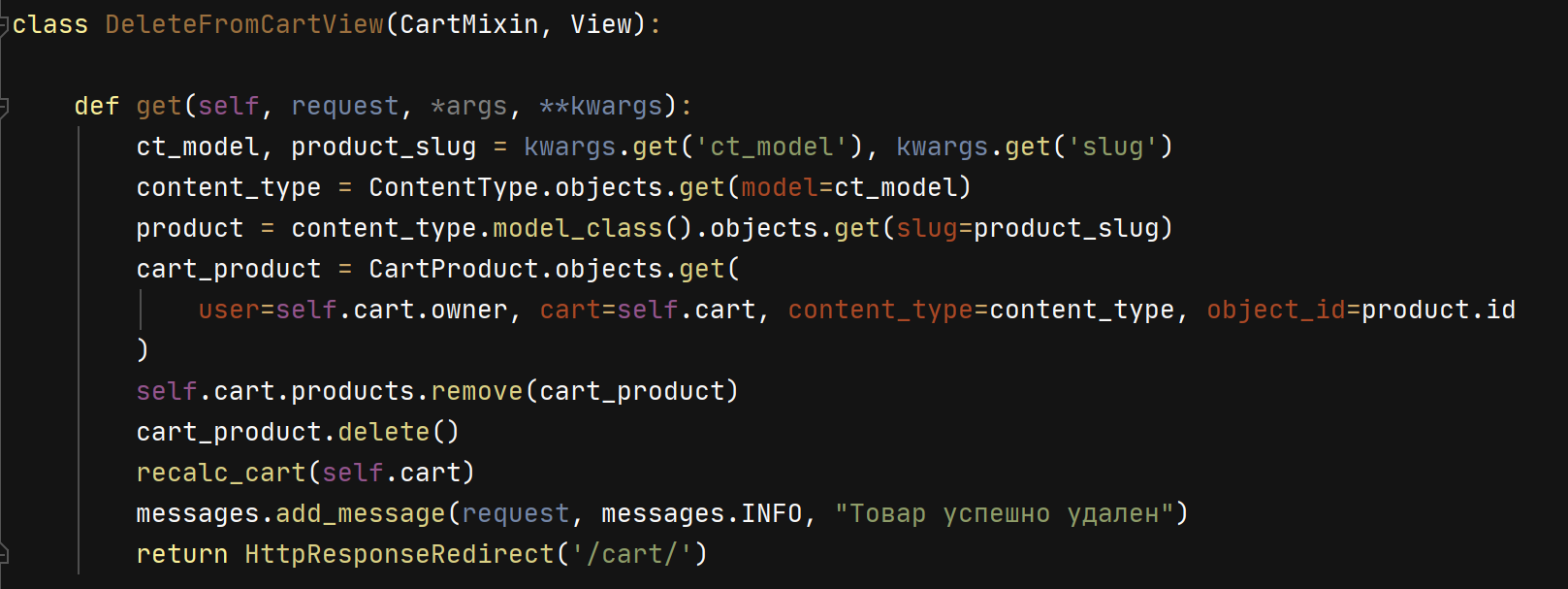


Рис.3.3.2 Пример использования Post запроса в программе

1. Пример работы сайта

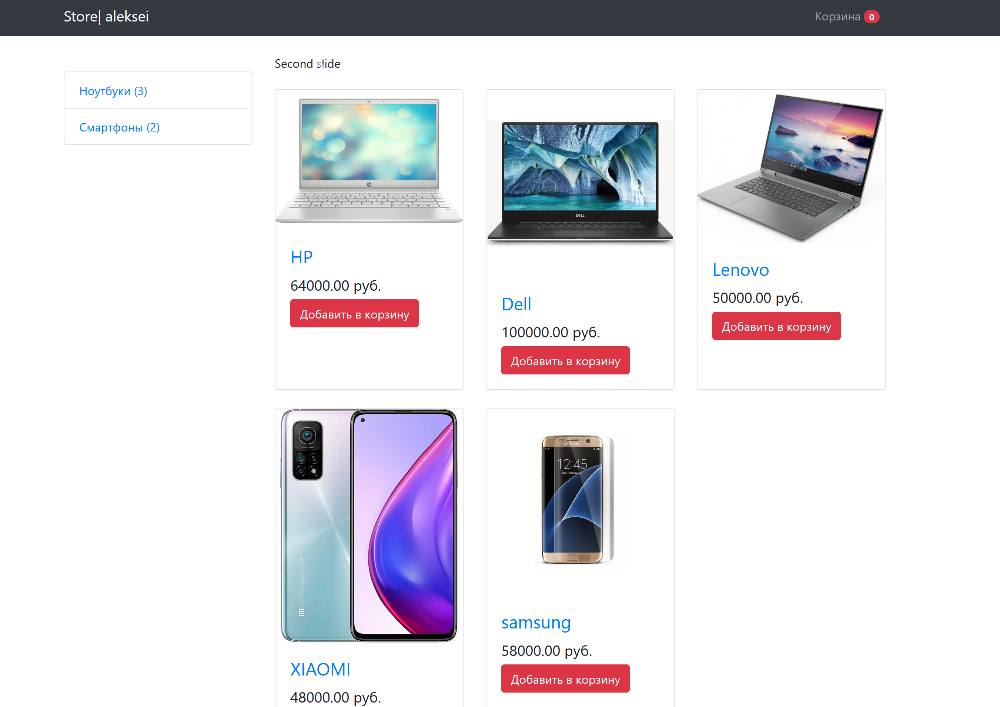


Рис. 4.1 Пример работы сайта с продуктами

Для проверки работы сайта было создано пять различных продуктов, три из которых входят в категорию ноутбуки, два – смартфоны.

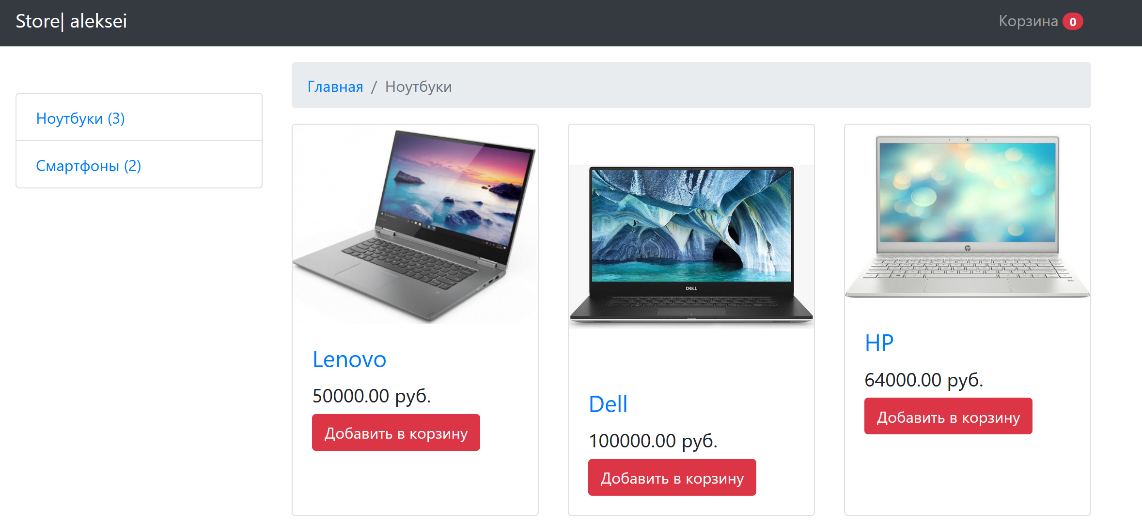


Рис. 4.2 Пример фильтрации

На сайте есть фильтр для выбора товаров по конкретным категориям.

1. Пример работы заказа

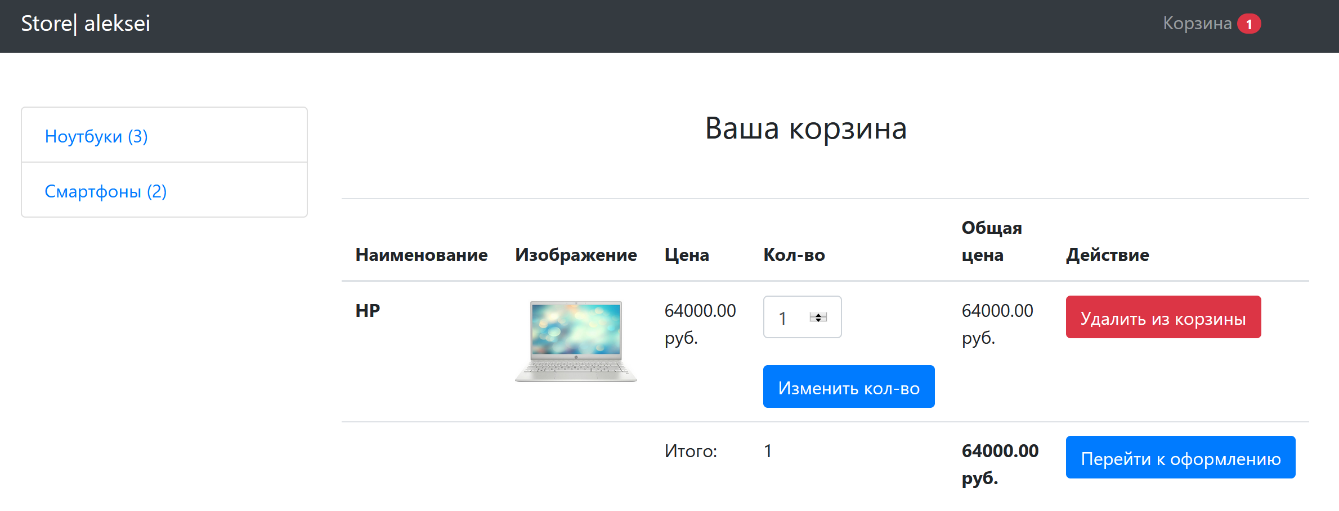


Рис. 5.1 Товар в корзине

Для добавления товара в корзину покупатель должен нажать “Добавить в корзину”, после этого он перейдет в корзину, а на значке корзины появиться число товаров в корзине.

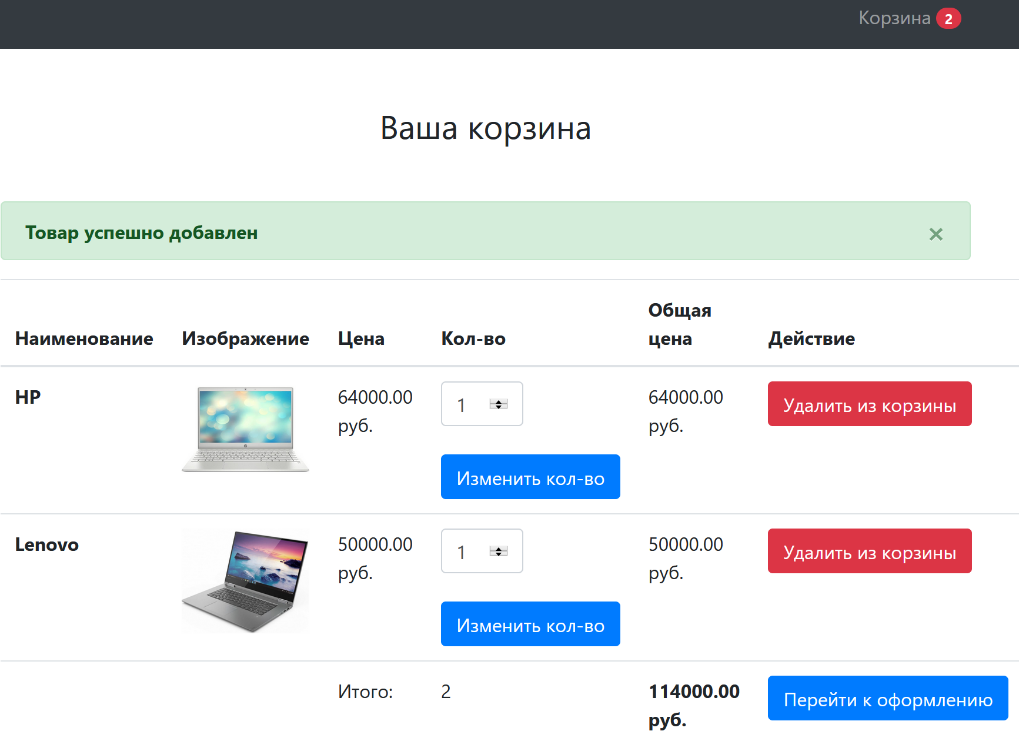


Рис. 5.2 Добавление товара в корзину

После добавление второго и более товаров в корзину покупатель будет видеть уведомление об успешном добавление товара в корзину.

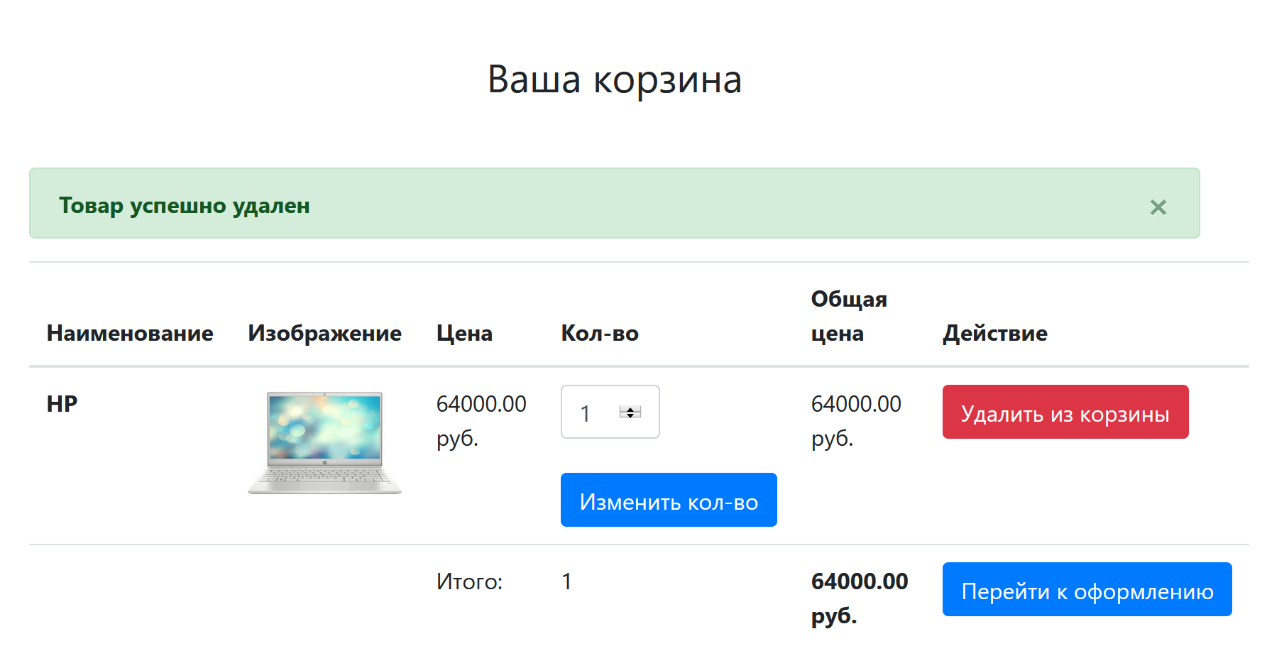


Рис. 5.3 Удаление товара из корзины

Для удаления товара из корзины достаточно нажать “Удалить из корзины”, после чего появится системное сообщение об успешном удаление.

1. Вывод

В ходе выполнения данной работы мы познакомились с фреймворком Django и узнали, что такое Post запросы. На практике применили знания, полученные на лекциях и в ходе выполнения данного задания.

Необходимо отметить, что это работа стала второй по написанию веб-приложения на локальном сервере. Первая была выполнена на JS на фреймворке React. Полученный нами опыт позволяет сравнить два подхода к разработке веб-приложений. На основе проведенного анализа сделан вывод о том, что фреймворк Django отличается сравнительной простотой в разработке, обладает мобильностью и широкими возможностями для быстрого развертывания веб-приложений.

1. Список литературы
2. https://metanit.com/python/django/
3. https://ru.stackoverflow.com/questions/
4. https://getbootstrap.com/docs/
5. https://www.djbook.ru/
6. https://djangobook.com
7. https://ru.wikipedia.org/wiki/Django
8. Swaroop C H "A Byte of Python".