

**ДОКУМЕНТАЦИЯ К ПРОЕКТУ**  
**E-Store (интернет-магазин техники)**

Выполнил: Соколов Иван Александрович

Группа: РПО 23/2

**Ссылки:**

Figma-макет: <https://www.figma.com/community/file/1362344995738653261/e-store-mobile-web>

Задания от научного руководителя: <http://exercise.develop.maximaster.ru/book/intro/>

Репозиторий содержит папки: backend, frontend, docs.

Документ размещается в папке docs.

## **1. Общие сведения**

Проект E-Store - учебный веб-сайт интернет-магазина (англоязычный интерфейс), созданный по открытому бесплатному макету из Figma. Сайт демонстрирует верстку и базовую интерактивность типичных страниц e-commerce, а также отдельную страницу с заданиями из практики.

Страница с заданиями отсутствовала в исходном макете. Она добавлена отдельно и доступна из шапки сайта по пункту меню «Задания» (единственная русскоязычная кнопка и подпись на сайте).

Для страниц, которые не реализованы в рамках практики, добавлены «заглушки» с сообщением о том, что раздел находится в разработке. Переключение между страницами выполнено через React Router.

## **2. Технологический стек**

Frontend:

- React + Vite + TypeScript
- SCSS (модульная стилизация компонентов)
- React Router (маршрутизация)
- Swiper (слайдеры для категорий и пагинации)
- react-chartjs-2 (график загруженности процессора в заданиях)
- Интеграция Яндекс.Карт (форма заказа в заданиях)

Backend:

- Django (API и административная панель)
- Django Admin (создание/редактирование категорий, товаров и их вариаций)

Хранение данных на клиенте:

- localStorage используется для сохранения состояния кнопки «избранное» и данных электронной таблицы (задания).

### **3. Структура репозитория**

Репозиторий организован в виде трех папок:

- backend - серверная часть на Django.
- frontend - клиентская часть на React/Vite.
- docs - документация и отчет по практике.

### **4. Запуск проекта на локальном компьютере (Windows, PowerShell)**

Ниже приведена типовая инструкция для запуска на другом ПК. Команды выполняются из PowerShell. Перед началом убедитесь, что установлены Git, Node.js (LTS) и Python 3.x, а также что они доступны в PATH.

#### **4.1. Клонирование репозитория**

Откройте PowerShell и выполните:

```
git clone <URL_ВАШЕГО_РЕПОЗИТОРИЯ>
cd <папка_репозитория>
```

#### **4.2. Запуск backend (Django)**

Перейдите в папку backend и создайте виртуальное окружение Python:

```
cd backend
python -m venv venv
```

Активируйте окружение:

```
.\venv\Scripts\Activate.ps1
```

Установите зависимости и выполните миграции базы данных:

```
pip install -r requirements.txt
python manage.py migrate
```

Создайте администратора (рекомендуется на новом ПК, если база данных создается «с нуля»):

```
python manage.py createsuperuser
```

Запустите сервер разработки:

```
python manage.py runserver
```

После запуска backend будет доступен по адресу <http://127.0.0.1:8000/>, административная панель - <http://127.0.0.1:8000/admin/>.

### 4.3. Запуск frontend (React + Vite)

Откройте второй терминал PowerShell, перейдите в папку frontend и установите зависимости:

```
cd frontend
```

```
npm install
```

Запустите фронтенд в режиме разработки:

```
npm run dev
```

После запуска frontend обычно доступен по адресу <http://localhost:5173/>. Frontend обращается к backend для получения данных товаров и категорий. Если порт backend изменен, адрес API нужно скорректировать в настройках фронта (vite.config.ts, src\shared\api\productsApi.ts).

## 5. Навигация и общий UI

Шапка (header) и подвал (footer) отображаются на всех страницах. Шапка закреплена в верхней части экрана и остается видимой при прокрутке.

Под шапкой реализована история переходов по страницам. Она отображает цепочку посещенных разделов и текущую страницу. Любой предыдущий элемент в истории кликабелен и возвращает пользователя назад без ручного поиска по меню.

## 6. Главная страница

Главная страница состоит из нескольких секций, повторяющих структуру макета.

## **6.1. Первая секция (Welcome)**

В первой секции размещена кнопка Shop Now. Это отдельный React-компонент, который принимает цвет как входной параметр. За счет этого один и тот же компонент используется в разных блоках сайта, но визуально может отличаться. Кнопка ведет на страницу каталога.

## **6.2. Вторая секция (Promo)**

Во второй секции представлены промо-карточки отдельных товаров. В одной из карточек также используется кнопка Shop Now.

## **6.3. Третья секция (Категории)**

Секция категорий показывает 6 карточек одновременно. Листание выполняется стрелками над блоком или свайпом мышью. Прокрутка реализована библиотекой Swiper. При выборе категории пользователь попадает в каталог, где отображаются только товары выбранной категории. Привязка товаров к категориям выполняется на backend через административную панель.

## **6.4. Четвертая секция (Мини-каталог)**

Мини-каталог содержит несколько вкладок. В идеальном варианте вкладки должны показывать разные наборы товаров, но из-за ограниченного количества исходных данных реализовано изменение порядка карточек при переключении вкладок.

## **6.5. Карточки товаров, популярные товары и секция «скидок»**

Карточка товара включает изображение, название, цену и кнопку Buy Now. По нажатию открывается страница товара, и содержимое страницы зависит от выбранной карточки (данные подгружаются динамически).

Для всех кнопок на сайте добавлены hover-эффекты с плавной анимацией (масштабирование и изменение прозрачности).

Секция популярных товаров и промо-блоки построены на одном шаблонном компоненте карточки: визуальные отличия достигаются передачей разных данных (текст, изображения, цвет фона).

Секция товаров «со скидкой» отображает 4 карточки. Функционал реальных скидок в рамках проекта не реализован: вместо этого блок выбирает случайные 4 товара из общего списка и отображает их как отдельную подборку.

## **6.6. Анимации**

На главной странице добавлены легкие анимации появления некоторых секций и компонентов (плавный выезд снизу), что делает интерфейс более «живым», не перегружая пользователя.

## **7. Каталог товаров**

Страница каталога включает блок фильтров и список карточек товаров.

### **7.1. Фильтры**

Левая панель фильтров реализована на уровне интерфейса: блоки можно раскрывать и сворачивать с плавной анимацией, а пункты можно отмечать галочками. При этом реальная фильтрация по этим пунктам не выполняется.

Над списком карточек расположен рабочий фильтр сортировки:

- По рейтингу - возвращает товары в исходный порядок (система рейтинга в проекте не реализована).
- По убыванию цены - сортирует карточки по цене от большей к меньшей.
- По возрастанию цены - сортирует карточки по цене от меньшей к большей.
- Набор отображаемых пунктов фильтра зависит от способа входа в каталог:
- Если перейти через Shop Now с главной страницы - показываются фильтры для всех категорий.

- Если перейти из секции категорий - показываются фильтры, относящиеся только к выбранной категории.

Пункты фильтра захардкожены (не управляются через админ-панель), так как функционал фактической фильтрации по ним не используется.

## 7.2. Пагинация карточек

В каталоге реализована постраничная навигация внутри секции товаров: на одной странице отображается не более 9 карточек. Если товаров больше, они распределяются по страницам 2, 3 и далее.

Если количество страниц не превышает 5, отображается стандартный набор кнопок, например: < 1 2 3 >.

Если страниц больше 5, панель выбора страниц превращается в горизонтальный список на базе Swiper и работает по следующему принципу:

- По умолчанию видны первые три страницы, затем кнопка «...», затем кнопка последней страницы (например: < 1 2 3 ... 10 >).
- Кнопка «...» является активной: при нажатии панель разворачивается и показывает все страницы подряд, сохраняя возможность прокрутки стрелками или колесом мыши.
- Пока «...» не раскрыто, стрелками прокручиваются только первые элементы (1-3), а «...» и последняя страница остаются закрепленными и всегда видимыми.

Такой подход сохраняет компактность пагинации при большом количестве страниц и при этом оставляет быстрый доступ к последней странице.

## 8. Страница товара

Страница товара открывается по кнопке Buy Now из карточки. Данные на странице зависят от выбранного товара и подгружаются динамически.

## **8.1. Галерея изображений и просмотр крупно**

Верхняя часть страницы включает одну большую (активную) фотографию и вертикальный список миниатюр слева.

При выборе миниатюры происходит «обмен»: выбранная миниатюра становится активной, а предыдущая активная фотография перемещается в список миниатюр. Активная миниатюра всегда располагается первой и отображается полностью непрозрачной, остальные миниатюры - полупрозрачные.

При клике по активной фотографии открывается увеличенный просмотр: изображение фиксируется по центру экрана, появляется белый полупрозрачный оверлей, блокируются прокрутка и клики по остальной странице. Закрытие выполняется кликом по оверлею.

## **8.2. Вариации товара (цвет и память)**

Товар может иметь несколько цветовых вариантов (например, iPhone 14 Purple и iPhone 14 Gold). При смене цвета обновляются изображения и название товара, при этом переход на другие страницы не выполняется.

Цвета и их вариации задаются через админ-панель: для каждого цвета используется HEX-код, который должен совпадать между списком доступных цветов и описанием конкретной вариации (важно вводить один и тот же код, включая символ #).

Также для товара задаются варианты памяти (до 4 значений). Часть вариантов может быть помечена как недоступная - такие варианты отображаются, но выбрать их нельзя. Активность/неактивность вариантов памяти настраивается через админ-панель.

### **8.3. Характеристики, описание и раскрывающиеся блоки**

Краткие характеристики и условия покупки выполнены в виде повторно используемых компонентов: один и тот же шаблон принимает разные данные (текст и иконки), что упрощает поддержку и развитие страницы.

Короткое описание (один абзац) автоматически сворачивается, если превышает заданную длину. Кнопка *more* раскрывает текст плавно до полного объема, а подпись кнопки меняется на *Turn*. Повторное нажатие снова сворачивает описание.

Кнопки *Add to Wishlist* и *Add to Card* присутствуют в интерфейсе, но не выполняют действий, так как корзина и полноценное избранное в проекте не реализованы.

### **8.4. Подробное описание товара (*Details*) и список характеристик**

Для каждого товара в админ-панели задается подробное описание: абзац текста и набор характеристик в формате «название - значение». Количество характеристик не ограничено - они выводятся как список (через *map*).

Значение характеристики может быть либо строкой, либо списком. Если передан список, элементы выводятся построчно. В админ-панели для ввода списка используется разделитель «;» - этот символ не отображается на странице и служит только для разбиения на элементы.

Если блок *Details* становится выше заданного ограничения (из-за большого текста или большого количества характеристик), он автоматически обрезается и внизу появляется кнопка *View More*. По нажатию блок разворачивается полностью, кнопка меняет подпись на *Turn* и при повторном клике снова сворачивает блок. Все переходы сопровождаются плавными анимациями.

## **9. Локальное «избранное» (сердечко на карточке)**

На карточке товара есть кнопка в виде сердечка. Полноценный раздел «избранное» не реализован, но состояние кнопки работает корректно: при нажатии сердечко становится красным с плавной анимацией и сохраняет состояние после обновления страницы.

Сохранение выполнено через localStorage. Для корректной привязки состояния каждому товару назначается уникальный id в админ-панели.

## **10. Страница «Задания» (практика)**

Раздел «Задания» содержит реализацию упражнений по материалам учебного курса (ссылка указана на титульной странице). Задания выполнены на JavaScript/TypeScript и стилистически приведены к общему оформлению сайта.

### **10.1. Случайный цвет прямоугольника**

Пользователь задает ширину и высоту прямоугольника в двух числовых полях. Кнопка «Случайный цвет» устанавливает прямоугольнику случайный цвет фона.

### **10.2. Оформление заказа + Яндекс.Карты**

Форма заказа включает обязательные поля ФИО, телефон и e-mail, поле комментария и интерактивную карту с меткой доставки.

Проверки формы:

- Телефон: разрешен ввод только цифр; максимальная длина - 11 символов. Если символов меньше, при отправке показывается ошибка.
- E-mail: при отправке проверяется корректная структура с символом @ (текст должен быть и до @, и после @). Пробелы запрещены.
- Комментарий: необязательное поле, ограничение - 500 символов.
- Компонент Яндекс.Карт позволяет перемещать карту и устанавливать метку места доставки.

### **10.3. Таблица товаров из JSON + фильтр по диапазону цен**

Данные товаров хранятся в JSON-файле внутри проекта. Таблица считывает данные и выводит их на экран.

В таблице отображаются количество товара и цена за единицу, а общая сумма рассчитывается автоматически как произведение количества на цену.

Над таблицей расположен фильтр диапазона цен за единицу. Товары вне диапазона скрываются. Если подходящих товаров нет, выводится сообщение об отсутствии результатов.

### **10.4. График загруженности процессора**

График получает значения загруженности из JSON-файла (реальные данные процессора нечитываются). Запросы выполняются каждые 5 секунд, после чего график обновляется. Визуализация реализована через react-chartjs-2.

Под графиком выводятся: количество выполненных запросов и процент ошибочных запросов.

### **10.5. Электронная таблица**

Таблица позволяет вводить произвольный текст в ячейки без ограничения длины. Кнопки + и - снизу и справа динамически добавляют/удаляют строки и столбцы. Если столбцов становится слишком много и таблица выходит за границы контейнера, ширина столбцов уменьшается, что позволяет продолжать добавление без поломки интерфейса.

Все введенные значения сохраняются в localStorage и остаются после обновления страницы или перезапуска сайта. Реализация выполнена без дополнительных библиотек.

## **10.6. Счетчик посещений (hits)**

Счетчик увеличивается на 1 при каждом обновлении страницы или повторном заходе пользователя на сайт. Функционал реализован на backend (Django).

## **11. Административная панель (Django Admin)**

Административная панель предназначена для управления контентом сайта: категориями, товарами, их вариациями, описаниями и характеристиками.

Данные для входа: логин admin, пароль A123\_A123. Если проект запускается на новом ПК с новой базой данных, учетную запись администратора нужно создать командой `python manage.py createsuperuser`.

### **11.1. Добавление товара (Products -> Products)**

- 1) В левой панели откройте вкладку Products и в самом низу выберите пункт Products. Откроется список уже созданных товаров.
- 2) Нажмите кнопку Add Product (в правом верхнем углу). Откроется форма создания товара.
- 3) Заполните поля в следующем порядке:

- id - уникальный ключ товара. Не должен повторяться, не должен содержать пробелов и специальных символов.
- title - название товара для карточки и страницы товара, например: Apple iPhone 14 Pro 128GB Deep Purple (MQ0G3).
- category - выбор категории из уже созданных (привязка товара к разделу каталога).
- Sale Price - актуальная цена (используется в карточке и на странице товара).
- Full Price - «старая» цена. Отображается как полупрозрачный перечеркнутый текст рядом с Sale Price на странице товара.
- preview image - изображение только для карточки товара. Файлы изображений находятся по пути `frontend/src/assets/images`.

- 4) Секция Images: содержит 4 поля для изображений. Заполнять все 4 не обязательно, но должно быть заполнено минимум одно поле.
- 5) Секция Description: введите абзац текста с описанием товара.
- 6) Секция Product memory options: добавьте варианты памяти (например, 128GB, 256GB и т.д.). Максимум - 4 варианта.
- 7) Секция Product disabled memory options: укажите варианты памяти, которые должны быть недоступны. Чтобы сделать вариант недоступным, введите тот же текст, который использовался в Product memory options (например, 128GB).
- 8) Секция Product color options: добавьте цвета товара. В каждое поле вводится HEX-код цвета с обязательным символом #. Нужно добавить минимум один цвет (базовый).
- 9) Секция Product color variants: добавьте вариации по цветам - изображения и названия для каждого цвета, например: Apple iPhone 14 Pro 1TB Gold (MQ2V3).
  - В поле color hex в этой секции нужно указать тот же HEX-код, который был добавлен в Product color options. Коды должны совпадать.
  - Для базового цвета также обязательно заполнить название и минимум одно изображение.
- 10) Секция Product characteristics: добавьте характеристики товара (рекомендуется минимум 3), например: Screen size 6.7". Для каждой характеристики выбирается иконка.
- 11) Секция Product terms: добавьте условия покупки (рекомендуется минимум 3), например: Free Delivery 1-2 day, и выберите иконку.
- 12) После заполнения нажмите Save.

## **11.2. Добавление подробного описания (Products -> Product description sections)**

- 1) В левой панели во вкладке Products выберите Product description sections.
- 2) Нажмите Add product description section.

3) Заполните форму:

- Product - выберите товар, к которому будет привязано описание (будет виден id и title).
- Details title - заголовок блока (например, Details или Description).
- Details desc - абзац текста с подробным описанием.

4) Секцию Product description characteristic groups можно не заполнять на этом шаге. Нажмите Save.

### **11.3. Добавление групп характеристик (Products -> Product description characteristic groups)**

1. В левой панели откройте Product description characteristic groups и нажмите Add.
2. В первом поле выберите section, к которой привязывается группа (секции нумеруются по возрастанию).
3. В поле Title задайте название группы, например Screen.
4. В секции Product description characteristic items добавьте характеристики в формате «название - значение», например: Screen diagonal - 6.7".
5. Если значение должно быть списком, вводите элементы через символ «;». На странице каждый элемент будет выведен с новой строки, а сам символ «;» не будет отображаться.
6. Нажмите Save.

### **11.4. Добавление категорий (Products -> Product categorys)**

1) В левой панели во вкладке Products выберите Product categorys и нажмите Add.

2) Заполните поля категории:

- id - уникальный идентификатор, не повторяется и не содержит пробелов/спецсимволов.
- title - название категории.

- icon - иконка категории.
- order - порядок отображения в свайпере (0 - первая, 1 - вторая, 2 - третья и т.д.).

3) Нажмите Save.

## 12. Итог

Проект E-Store демонстрирует связку React/Vite (frontend) и Django (backend) с управлением контентом через админ-панель, динамическими страницами каталога и товара, а также отдельным разделом с заданиями из практики. Реализованы важные UI-механики: роутинг, слайдеры Swiper, сохранение состояния в localStorage, постраничная навигация и интерактивная галерея на странице товара.