**Лот №2. Разработка frontend**

**Альтернативная реализация раздела «Интернет-магазин / Смартфоны»**

На действующем сайте есть раздел «Интернет-магазин / Смартфоны», отображающий каталог смартфонов и фильтрацию по нему. Текущую реализацию можно найти по ссылке <https://moskva.beeline.ru/shop/catalog/telefony/smartfony/>.

В рамках тестового задания необходимо реализовать функциональность каталога смартфонов с возможностью фильтрации и пагинации.

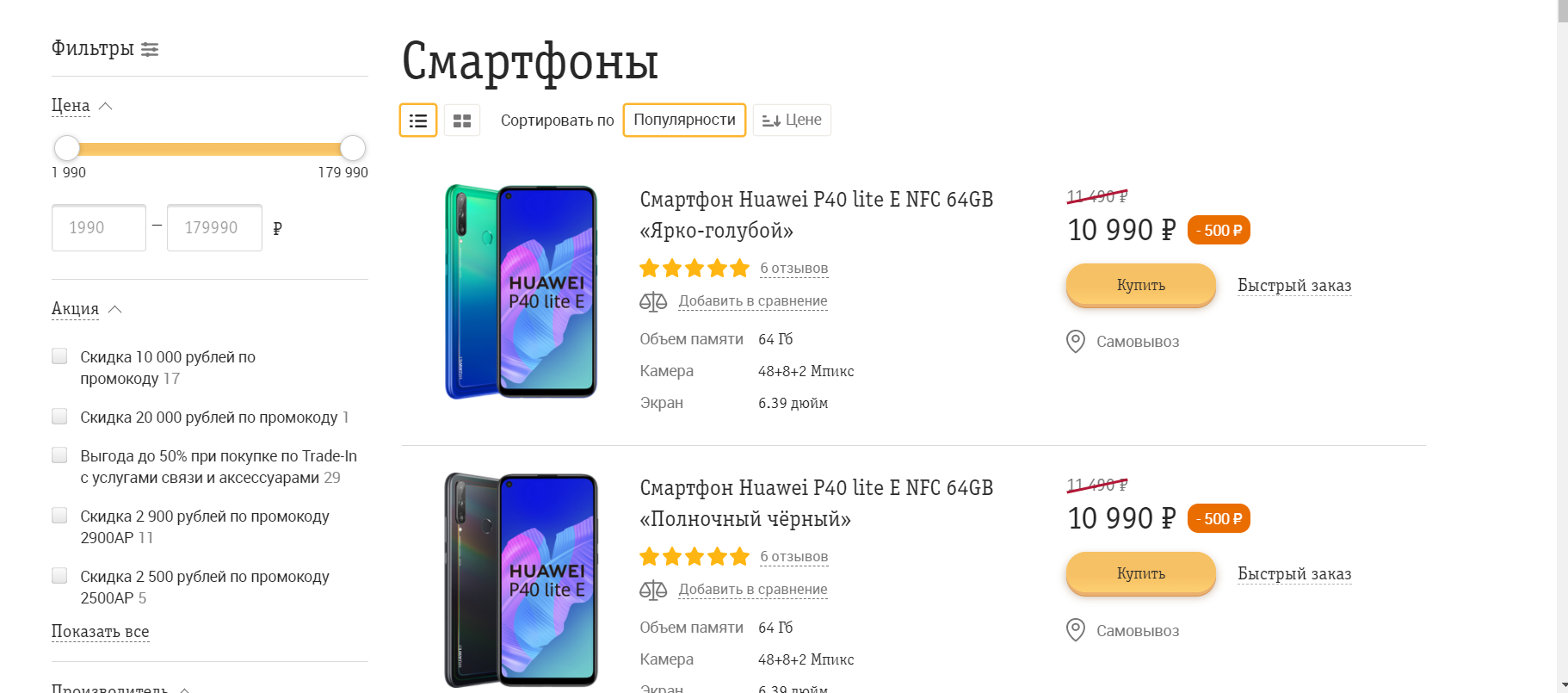


Рис. 1 Каталог смартфонов

В рамках тестового задания **не** нужно реализовать общую шапку и футер сайта. Сверху приложение должно выглядеть, как на рисунке 1. Внизу приложение должно заканчиваться пагинацией, как на рисунке 2.



Рис. 2 Пагинация каталога

Приложение должно реализовывать следующие функциональность и элементы:

1. Каталог смартфонов с возможностью представления в двух видах: списком и плитками.
2. Возможность сортировки смартфонов по популярности и цене.



1. Возможность фильтрации смартфонов по параметрам: цена, акция, цвет, объем оперативной памяти. Указанные фильтры должны выглядеть, как на исходном ресурсе. Остальные фильтры реализовывать не надо.

1. Карточка смартфона. Фотографию взять произвольную из интернета. Рейтинг (звезды) будет заполняться по полю объекта смартфона. На кнопки «Купить», «Добавить в сравнение» и «Быстрый заказ» повесить заглушки. По нажатию на отзывы должно открываться модальное окно со списком отзывов.
2. С учетом отсутствия API необходимо проработать модели и интерфейсы всех необходимых сущностей и сгенерировать тестовые данные, эмулировать работу с API с учетом пагинации, фильтрации и сортировки через API.

Необходимо проанализировать и подготовить основные пользовательские сценарии, сформулировать требования к системе, соизмерить их с коммерческими задачами телеком-оператора и изложить замеченные плюсы и минусы.

Далее, необходимо предложить улучшенный собственный вариант решения сформулированной задачи.

Технологические требования к приложению:

* В качестве языка программирования использовать TypeScript.
* В качестве хранилища состояния использовать Redux.
* Приложение должно работать на десктопе, мобильная версия не нужна. Минимальное разрешение экрана – 1280х1024 px.
* Приложение должно работать с современными версиями Chrome и Firefox. Будет большим плюсом возможность работы через IE11.

Ожидаемый материальный результат (минимум):

* Анализ плюсов и минусов текущей версии раздела
* Разработка требований и ТЗ системы
* Работоспособный прототип (с учетом заглушек и тестовых данных)
* Документация (описание решения, инструкция по сборке, инструкция по установке)
* Исходные коды приложения, пригодные для самостоятельной сборки дистрибутива

**​**