**По критериями оценки:**

1. Динамичность веб-страницы:

- <http://localhost:3000/catalog/page-1> - список товаров, которые можно сортировать. У каждого товара сделать отметку Favorite. При данной отметке появляется счетчик отмеченных как Favorite товаров + отметке галочки у счетчика – Favorite подсвечиваются цветом. Кнопка «Добавить в корзину» заблокирована, если у товара цена 0, НО расблокируется, если отметить как Favorite.

При клике «Добавить в корзину» появляется/скрывается уведомление. Может быть разным, в зависимости от того, если ли этот товар в корзине уже или нет.

- <http://localhost:3000/basket> – страница корзины. Поля формы с валидацией.

При загрузке обоих страниц используется GIF пока данные загружаются.

1. Производительность отрисовки:

- <http://localhost:3000/catalog/page-1> – 50 динамичных однородных элементов.

- <http://localhost:3000/basket> – анимация при удалении товаров из корзины.

1. Навигация в приложении:

-реализован роутинг + на странице <http://localhost:3000/basket> – уведомление о потере данных.

- <http://localhost:3000/catalog/page-1> – постраничная навигация.

1. Кроссбраузерность:

- проверена в разных браузерах на одном компьютере. К сожалению, не было возможности проверить хотя бы на одном мобильном устройстве.

- Адаптивная верстка под узкий экран.

1. Коммуникации:

-Используется AJAX для загрузки и получения данных со списком товаров и списком товаров в корзине. GIF анимация при загрузке данных.

1. Модель данных:

- json-Server + Redux

1. Cборка проекта:

- Живая DEV-сборка: npm start + npm run mock:api (на 3004 порту для json-сервера)

- npm run build: serve -s build + npm run mock:api – у меня все работает )

1. Тесты – БЕДА. Есть тесты отдельных функциональных возможностей. Если по остальному функционалу все ОК, буду пытаться еще дописать тестов до 07.12.

**Комментарии по коду компонентов:**

**PageMain** – компонент в функциональном стиле. Рендерит просто текстовую информацию.

**PageCatalog** – компонент, который связан с Redux Store и получает в качестве пропсов каталог товаров и содержимое корзины из Redux. Здесь организована отправка данных в Redux и сохранение в json файл.

Здесь рендерятся компоненты ShopCatalog, ShopCatalogItem и Success.

**ShopCatalogItem** - компонент, который рендерит информацию по отдельному товару и отображаться по-разному, в зависимости от ShowMode (CATALOG\_ITEM - вид при отображении в списке товаров, SINGLE\_ITEM - отдельная страница товара). Передача данных о клике по кнопке добавления в корзину или Favorite происходит с помощью EventEmitter.

**ShopCatalog**: рендерит компоненты SortingSelect, FavoriteCount, Pagiation и ShopCatalogItem. Подписывается на эмиты о сортировке и отметке Favorite. Реализованные методы:

-colorAllFavoriteProducts - закрашивает отмеченные как Favorite товары при выбранном checkbox checked

-onPageChanged - кол-бэк от компонента Pagination. Передаются данные для формирования списка товаров на выбранной странице.

-getSortedProductsByName, getSortedProductsByPrice, getSortedProductsArray - методы, отвечающие за сортировку товаров в зависимости от выбранного варианта. Плюс вызывается метод, связывающий с пагинацией. Чтобы при сортировке пагинация работала верно.

**SortingSelect** - select с вариантами сортировки списка товаров. При сортировке в url добавляется префикс, в зависимости от выбранного варианта. С помощью emit передается информация о том, что сделан какой то выбор.

**Pagination** - компонент, отвечающий за постраничную навигацию. Здесь происходит подсчет общего количества страниц, в зависимости от лимита отображения на странице. По умолчанию 50 наименований на странице.

**FavoriteCount** - компонент, отвечающий за подсчет количества товаров, отмеченных как Favorite. Отображается, если отмечен хотя бы один, иначе скрыт. Рендерит Checkbox, при выборе которого товары Favorite закрашиваются цветом.

**ShopCatalogItem** - из ShopCatalog рендерится в режиме showMode: CATALOG\_ITEM.

Если цена 0 - кнопка "положить в корзину" заблокирована. Если при этом сделать отметку Favorite - тогда товар можно отправить в корзину.

При отметке Favorite данные сохраняются только в Redux (при переходе по страницам + back и forvard в браузере - отметка Favorite и счетчик Favorite не сбрасываются). Сброс Favorite при перезагрузке страницы.

**PageBasket** - Компонент отвечающий за корзину товаров. Получает в качестве пропсов список товаров в корзине из Redux.

Рендерит компонент ShoppingCart, который в свою очередь ренедрит ShoppingCartItem и Checkout.

**ShoppingCart**: рендерит список товаров, которые находятся в корзине. Здесь же методы добавления и удаления из корзины. Все данные сохраняются в Redux и json. Здесь реализована анимация при удалении.

**Checkout**: миниформа заказа. Реализована валидация, если хотя бы одно поле не заполнено. При нажатии на кнопку с незаполненными полями появляется уведомление, о том, что поле обязательно для заполнения. После валидного заполнения и клика по кнопке Send - происходит удаление всех товаров из корзины + редирект на страницу заглушку, которая рендерится компонентом Success. Данные из полей формы никуда не отправляются.

Если поля были заполнены, но происходить переход на другую страницу - появляется предупреждение, что данные в полях не сохранятся.

**Alerts** – компонент, отвечающий за появление уведомления, при добавлении товаров в корзину. Товар в корзину можно добавить только 1 раз. При повторной попытке отправить товар в корзину, показывается уведомление в красном фоне. Рендерится из App.js

В папке api – db.json файл со списком товаров.

В папке src/Redux – в отдельных папках Actions, Reducers, Thunk.