Компанент Header- отображает ссылки между переходом в карзину товаров и обратно на домашнюю страницу HomePage(карзина товаром содержит функцию cartCount() отображает количество товаров в ней)

Компанент ProductCard (Карточка продукта) отрисовывает каждый товар на домашней странице HomePage с кнопкой (Добавить товар в карзину) отвечает за это функция addToCart()

И этот компанент вкладываем в Компанент HomePage(Домашняя страница) он имеет HOOK useEffect-который принемает асинхронную функцию через dispatch()

И имеет не большое условие при задержки загрузки (показывает сообщение Загрузка товаров) или показывает ошибку Error, это если товар по каким либо причинам не загрузился на страницу...

useSelector дёргает из state список продуктов

Компаненнт CartPage(Страница Карзины) отрисовывает карзину с тернарм условием, если она пустая сообщает, что Ваша карзина пустая,

если нет map-ит добавленный в неё товар с предварительной суммой каждого товара, так же в компаненте CartPage отображается итоговая сумму товаров, с помощью функции calculateTotal(),

Далле, сам компанент CartPage(Страница Карзины) принимает в себя Компанент CartItemComponent(каждый добавленный элемент в карзину) map-ится упорядочным списком ul

Сам Компанент CartItemComponent(элемент корзины) с вою очередь отрисовывает каждый добавленный товар, с функциями через диспечеры принимающие в себе редьюссеры (updateQuantity и removeItem)

productIncrease(увеличение количества товара и показывает предварительную сумму текущего товара) и

функцию productDecrease(уменьшить количество товара) ну и кнопка удалить с функцией remove(удалить товар)

Далее...

Весь нашь контент в Компоненте App(Апликейшен) оборачиваем в <Provider и передаём ему наш store=p{store}>

В store.ts создаём объект store функцией configureStore-внутри неё создаётся два reducer-ра... сначало productsReducer(в нутри которого с помошью productsSlice-а и extraReducers-ов при запросе функция createAsyncThunk сначало устанавливаем изначальное состояние пустой [] и потом получаем наши данные produrts )

(с тремя состояниями case-ми- pending, fulfilled и rejected)

Потом в этом же store-е создаём редьюсер cartReducer в нём создаём cartSlice - в котором имеются редьюссеры для обработки продуктов в карзине,

прибовлять, умножать, уменьшать и удалять-это addItem-Добавить элемент-продукт в компаненте(ProductCard), updateQuantity-Для добавления, именьшения и умножения количества товара в компаненте (CartItemComponent в карзине покупок),

removeItem-для удаления товара в этом же компаненте (CartItemComponent)

в api.ts лежит отдельно асинхронная функция-Получение моковых данных о продукте с помощью fetch запроса (с сайта mocki.io)