ПАПКА **COMPONENTS:**

1) В компоненте **Table**, используется библиотека ‘react-table’, для создания таблиц :

Именно в этом компоненте **Table** где-то 40% функции взяты из офиц документации для корректной работы самой таблицы)

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Функция **fuzzyTextFilterFn** предназначена для поиска по таблице, который будет находить строки, содержащие частичное соответствие введенному значению.

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описаниеФункция **filterTypes** фильтрует наши строки.

Функция **getTrProps** Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание и **rowProps** предназначены для настройки свойств строк таблицы.

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, программное обеспечение

Автоматически созданное описание

Функция **handleDeleteSelectedRows** вызывается при клике кнопки “Удалить”

В параметрах получаем объект event, где содерится вся информация о событии.

**selectedFlatRows** - это массив отмеченных наших строк в таблице, получается я пробегаю по этому массиву и возвращаю новый массив уже не со всеми данными о строке, а только id не строки, а именно что переданные данные в строке

Передаю данные уже в **onDelete –** внутри него уже делается запрос на удаления данных. Аргумент **pages** передается на какой странице была нажата кнопка удалить

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Функция **handleUserSelectedRows** – она срабатывает при клике кнопки добавить именно для страницы “Пользователи”**.** Эта функция предназначена для того чтобы добавлять роли выбранному пользователю, также идет проверка сколько выбрано строк, можно сказать мини валидация)

2) Компонент **RegistrationForm**

Изображение выглядит как текст, снимок экрана, программное обеспечение, Шрифт

Автоматически созданное описание

Данная функция **registration** отвечает за процесс регистрации нового пользователя. В 41 строчке я получаю роль(объект) по значению USER

В 42-48 передаю email пользователя, пароль, и массив ролей. Если регистрация прошла успешно, функция сохраняет токен, полученный в ответе на запрос, в локальном хранилище браузера. Потом сразу переходим на форму авторизации navigate('/')

3) Компонент **AuthorizationForm**

Изображение выглядит как текст, снимок экрана, программное обеспечение

Автоматически созданное описание

В 39-42 идет получение ответ от сервера с токеном доступа. Если ответ получен успешно, то функция сохраняет токен в локальном хранилище и диспатчит два экшена: setAuthMenu и setUser, которые устанавливают флаг авторизации пользователя и сохраняют его данные.

Затем идет проверка на то, что пользователь админ или нет

4) Компонент **ModalAddProduct**

Изображение выглядит как текст, снимок экрана

Автоматически созданное описаниеСрабатывает при сабмите формы, отправляет новый продукт на сервак

Изображение выглядит как текст, снимок экрана, Шрифт, программное обеспечение

Автоматически созданное описание

Происходит сохранения данных в redux

Изображение выглядит как текст, снимок экрана, дисплей, программное обеспечение

Автоматически созданное описание

Сабмитится уже для формы из страниц брендов или категорий

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Эти функции предназначены для

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

5) Компонент **Menu**

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Выход из системы

6) Папка **filters**

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описаниекомпоненты из react-table

7) Компонент **ListProduct**

Изображение выглядит как текст, снимок экрана

Автоматически созданное описаниеВ это длиннющем useEffect(хук жизненного цикла, который срабатывает при каждом рендере)

Происходит добавление данных в таблицу

Изображение выглядит как текст, снимок экрана

Автоматически созданное описание

8) Компонент **App**

**Изображение выглядит как текст, снимок экрана, программное обеспечение

Автоматически созданное описание**

Эта функция **chetAut** выполняет запрос на сервер для обновления токена доступа (access token) пользователя, если старый токен истек или не является действительным.

Компоненты **ListBrands**, **ListCategories**, **ListUsers**, такие как **ListProduct**