

Тестовое задание

Требования к реализации:

1. Использовать [Fake Store API](#)
2. Разработать веб-интерфейс:
 1. Для сущности "Products"
 1. Каталог продуктов
 2. Страница одного продукта
 3. Форма создания продукта
 2. Авторизация пользователя
3. Реализовать запросы (GET, POST, PUT, DELETE)
 1. Controllers
 2. Hooks (Ваши хуки должны возвращать результат хуков TanstackQuery возвращающих результат контроллеров)
 1. useProducts, useProduct для GET запросов
 2. usePostProduct, usePutProduct, useDeleteProduct для POST, PUT, DELETE запросов
 1. т.к. эти запросы не внесут изменения в Fake Store API, перезаписываем кэш на клиенте
4. Реализовать авторизацию пользователя
 1. Записать полученный token из API в cookie
 2. Хранить состояние isAuth
 1. Проверка наличия "token" в cookie при первой загрузке приложения (подготовить AuthProvider)
5. Использовать stack-технологий:
 1. React / Next.js
 2. Redux Toolkit / Zustand (храним только isAuth)
 3. Axios
 4. Typescript
 5. Tanstack Query
 6. React Hook Form
 7. Yup / Joi / Zod или любую другую библиотеку для валидации полей
 8. Любую UI-библиотеку на ваше усмотрение
6. Сделать публичный репозиторий на Github
7. Развернуть проект на Vercel