Технологии:
React / Next.js, TypeScript (only TS), ANTD, Storybook SCSS modules, Context (or Redux with RTKQuery, React-query).

Example: https://coinmarketcap.com/

API: Coincap
ICONS:
https://assets.coincap.io/assets/icons/btc@2x.png
https://assets.coincap.io/assets/icons/${...}@2x.png

Имена коммитов должны быть осмысленными.
Общий желательный дедлайн: 5 дней.

Задание 1: Таблица Монет

Создайте страницу, на которой будет отображаться таблица с информацией о различных монетах криптовалюты.

Ваша таблица должна включать следующие столбцы:

- *Символ монеты (например, BTC, ETH)*
- *Логотип монеты*
- *Цена в USD*
- *Рыночная капитализация в USD*
- *Изменение цены за 24 часа в процентах*
- *Столбец для кнопки **Add** для добавления монеты в портфель пользователя*

Требования:

- *Форматировать значения цены (b, m, k), максимальное количество знаков после запятой 2*
- *В таблице не должно быть нулевых значений (0.00$, 0, -)*
- *Должна быть пагинация данных с API (данные приходят частями)*
- *В таблице должны быть сортировки по цене монет, рыночной капитализации и изменения за 24 часа*
- *Поиск по названию монеты*

Задание 2: Страница с информацией о монете

Создайте страницу, на которой будет отображаться подробная информация об отдельной монете криптовалюты. Переход на эту страницу осуществляется при клике на строку монеты в таблице.

Ваша страница должна включать следующую информацию:

- Изображение монеты
- Название монеты
- Символ монеты
- rank
- Supply
- Цена в USD
- Рыночная капитализация в USD
- maxSupply
- График изменения цены за день/12 часов/1час - через выбор опции
- Кнопка для Add добавления этой монеты в портфель пользователя
- Кнопка вернуться назад для переход к Таблице

Требования:

- Обеспечьте хорошую визуальную структуру для отображения графика изменения цены.
- Показывать лоадер пока загружаются данные
- При неверном id монеты показывать сообщение об ошибке пользователю

Задание 3: Header и Портфель монет пользователя

В Header приложения отображается курс трех популярных криптовалют. Этот блок содержит информацию о цене каждой криптовалюты в USD.

Отдельный блок справа стоимость портфеля пользователя и разница с изначальной стоимостью портфеля, в скобках разница в процентах.
Example: 134,32 USD +2,38 (1,80 %).
При клике на этот блок открывается модальное окно где пользователь видит весь список своих монет, может удалять их из списка.

Стоимость портфеля пользователя:

- При клике на кнопку 'Add' откроется модальное окно с возможностью покупки n-числа монет.
- Сумма количества всех монет, умноженная на их цену в момент покупки, образует стоимость портфеля.
- Разница в стоимости - это разница между ценой портфеля с добавленными монетами на момент покупки и ценой этих монет, которая поступает в данный момент из API. При следующем запуске (после

релоада) приложения мы получаем текущие стоимости валют и можем увидеть разницу.
- Данные о портфеле пользователя хранятся в localStorage.
- Добавление монет должно подчиняться валидации на максимальное и минимальное количество.

Задание 4.

Добавить Storybook в проект и истории для каждого компонента в приложении.

Задание 5.

Покрыть playwright тестами основную логику фильтрация поиск, добавление/удаление, открытие модальных окон, валидация ошибок монет/расчет суммы и т.д.

Задание 6: Деплой результатов

Результаты вашей работы вы деплоите на любой бесплатный хостинг и ссылку отправляете нам на почту и закрепляете у себя в репозитории.
(Netlify, Vercel, Firebase, Github …)

Общие требования:

- **Задача должна выполняться как для реального заказчика**
- Ваше приложение должно быть адаптивным и должно хорошо выглядеть как на десктопе, так и на мобильных устройствах (адаптивность)
- Используйте TypeScript для типизации данных и компонентов.
- Обеспечьте чистоту кода, хорошую организацию и разделение компонентов.
- Вся описанная функциональность и требования должны быть реализованы и протестированы вами перед отправлением.
- Модальные окна закрываются при клике на оверлей
- Компоненты **Button, Input, Modal…** должны быть вынесены и переиспользуемыми.
- Не использовать bootstrap.

Оценка будет основана на:

- Функциональности и правильной реализации требований. (Частичная реализация не пройдет)

- Качестве кода и его чистоте (избегать ts-ignore, any типов и использовать prettier)
- Качественном и продуманном интерфейсе приложения.
- Ответственности при соблюдении требований и условий ТЗ.

Удачи при выполнении задания!

P.S Fullstack.
required - Replace api with Next.js, Tailwind, Mongoose, Mongodb.