Тестовое задание

Написать приложение, используя фреймворк React.js и язык Typescript. Разместить приложение в репозитории на GitHub с описанием в файле README.md на английском языке, как его запускать и использовать. Плюсом будет если вы выложите приложение на бесплатном хостинге (GitHub pages, Heroku) и предоставите ссылку в Readme.

Часть 1. Общая структура приложения

Приложение должно содержать:
— два раздела, переключение между которыми осуществляется через меню в шапке или раскрывающееся мобильное меню,
— страницу для ошибки 404.

Мобильное меню должно появляться только на экранах соответствующего размера.

Верстка должна состоять из трех корневых блоков — Header, Main, Footer. Дочерних блоков может быть сколько угодно.
— Header должен содержать кнопку для открытия/закрытия мобильного меню и само меню.

— Footer может быть заполнен данными об имени разработчика/контактах/копирайте и т.д.
— В Main — основной контент разделов. Контент в разделах может быть различным по объему.

Важно решить следующие вопросы:
1) Как сделать, чтобы Footer на странице с небольшим количеством контента «прилипал» к нижней границе окна?
2) Как отцентрировать содержимое и по вертикали, и по горизонтали на странице с ошибкой 404?
3) Как реализовать механизм открытия/закрытия мобильного меню?

Часть 2: Загрузчик изображений

Первый раздел содержит в себе форму, которая состоит из одного поля для ввода и кнопки подтверждения.
В поле можно ввести любой URL адрес на изображение, доступное в интернете (реализация механизма валидации URL адреса является опциональной).

После подтверждения формы изображение добавляется в список, который отображается ниже. Изображения в списке отображаются в виде квадратных карточек (как в Instagram) вне зависимости от размеров загружаемого изображения (соотношения ширины и высоты), по три карточки в строке. Изображение не должно быть искажено, оно должно замостить квадрат оптимальным образом (прямоугольное изображение должно быть обрезано до квадрата). Появление карточки должно происходить сразу после подтверждения формы, на время загрузки содержать изображение-placeholder и не изменять своих размеров в результате загрузки.

Добавлять карточки в список можно не ограниченное количество раз. Реализация механизма удаления карточек из списка является опциональной.

В этом разделе важным является корректное отображение карточек.

При реализации этой части задания вы можете столкнуться с проблемой загрузки изображений со сторонних серверов, вызванной политикой CORS, не нужно пытаться ее решить, либо ищите источники изображений где нет этого ограничения, либо загружайте изображения со своего локального сервера.

Часть 3: Клиент для произвольно выбранного внешнего API

Выберите какое-либо API из множества доступных в интернете (большим плюсом будет, если это GraphQL API).
Разместите форму, состоящую из одного поля, куда можно ввести какой- либо ключ, и кнопки

подтверждения, после нажатия на которую приложение в асинхронном режиме отправляет запрос к API и в результате выводит на страницу какие-либо данные.

Например, используя GitHub API, вводя имя аккаунта (например facebook, airbnb или ваш личный), вывести список открытых репозиториев (или какое-то их ограниченное количество).
Важно, чтобы состояние загрузки данных было очевидно для пользователя (например, показывался spinner).

NB! Стоит обратить внимание на оформление элементов. Проявите свои навыки дизайнера, знание CSS и HTML для оформления блоков, меню, кнопок, карточек, элементов списка. Допускается применение таких фреймворков как Bootstrap, Material UI и подобных.

При оценке работы внимание будет обращено на умение логично рефакторить код в соответствии с его предназначением, очевидно именовать коммиты, переменные, функции, компоненты и прочие сущности.

Наличие множества комментариев в коде не поощряется. Код должен быть понятен сам по себе.