**Тестовое задание на позицию стажёра-фронтендера**

Нужно разработать интерфейс для сайта [Hacker News](https://news.ycombinator.com/news), состоящий из двух страниц.

**Продуктовые требования**

**Главная страница**

* Показывает последние 100 новостей в виде списка, отсортированного по дате, самые свежие сверху.
* Каждая новость содержит:
  + название
  + рейтинг
  + ник автора
  + дату публикации
* По клику на новость происходит переход на страницу новости
* Список новостей должен автоматически обновляться раз в минуту без участия пользователя
* На странице должна быть кнопка для принудительного обновления списка новостей

**Страница новости**

* Должна содержать:
  + ссылку на новость
  + заголовок новости
  + дату
  + автора
  + счётчик количества комментариев
  + список комментариев в виде дерева
* Корневые комментарии подгружаются сразу же при входе на страницу, вложенные - по клику на корневой
* На странице должна быть кнопка для принудительного обновления списка комментариев
* На странице должна быть кнопка для возврата к списку новостей

**Технические требования**

* Приложение разработано с использованием React и Redux
* Использован [официальный API Hacker News](https://github.com/HackerNews/API). Вызовы Hacker News API и обработка данных от него производятся напрямую с фронтенда (кроме случая, если вы сделаете опциональное задание про Node.JS)
* Роутинг выполнен с использованием [React Router v5](https://github.com/ReactTraining/react-router/releases/tag/v5.0.0)
* Фреймворк UI любой на ваше усмотрение (как пример [Ant Design](https://ant.design/) или [Semantic UI](https://react.semantic-ui.com/)).
  + Можно и на чистом css, главное, чтобы было красиво
* Пакетный менеджер yarn
* Приложение должно запускаться по адресу localhost:3000 командой yarn start
* При переходах по ссылкам страница не перезагружается
* Исходный код решения должен быть выложен с вашего аккаунта на [Github](http://github.com/) с Readme файлом с инструкцией по запуску

**Опциональные задания**

* Использование TypeScript
* Бэкенд для хостинга статики и API для инкапсуляции внешних запросов на Node.JS
* Покрытие кода юнит-тестами