

# Тестовое задание для стажера

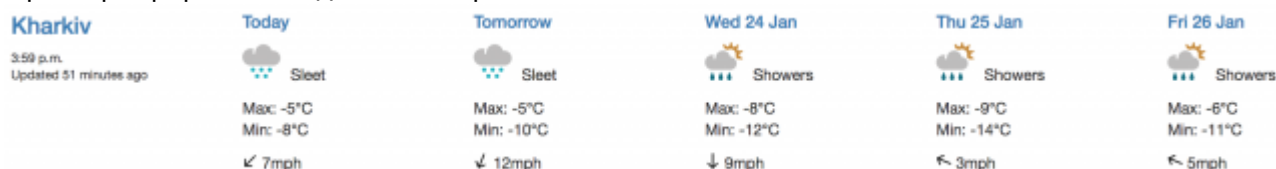
Теги: Web, Frontend, JavaScript, React

## 1. Общее описание

Необходимо создать одностраничное клиентское приложение для просмотра прогноза погоды с использованием библиотеки React.js. Приложение должно использовать публичный [API MetaWeather](#) для получения данных о прогнозе погоды в разных городах.

## 2. Описание экранов

1. Стартовый экран приложения - "Поиск". Основные элементы: текстовое поле ввода и список. Вверху экрана присутствует навигационное меню, позволяющее переходить между экранами "Поиск" и "Избранное". В поле вводится название или часть названия города латиницей. В списке отображаются результаты поиска городов по названию с использованием предоставленного API. Напротив каждого города в списке есть кнопка для добавления этого города в список избранных городов. Если город уже в избранном, кнопка неактивна.
2. Экран "Избранное". Основные элементы: текстовое поле ввода и список. Вверху экрана присутствует навигационное меню, позволяющее переходить между экранами "Поиск" и "Избранное". В поле вводится название или часть названия города латиницей. В списке отображаются результаты локального поиска среди городов, ранее добавленных в избранное. Если поле ввода пустое - отображаются все города из списка избранных. Напротив каждого города в списке есть кнопка для удаления этого города из списка избранных городов.
3. Экран "Прогноз". При клике на город из любого списка происходит переход на экран с пятидневным прогнозом погоды в этом городе, полученным по предоставленному API. Пример оформления данного экрана:



## 3. Другие требования и положения

1. Срок реализации: 3 дня.
2. Предоставленное API не использует HTTP заголовки для кроссдоменных запросов. Для решения этой проблемы предлагается использовать проксирование запросов к API, например, при помощи средств, предлагаемых шаблоном create-react-app.
3. Приветствуется, но необязательно использование вспомогательных библиотек, например: redux, redux-thunk, redux-saga, styled-components и др.
4. Качество выполненной функциональности оценивается выше количества.
5. Внешний вид (интерфейс) оценивается в последнюю очередь, но играет роль при прочих равных условиях.
6. Список избранных городов должен сохраняться после перезагрузки приложения (страницы).
7. Опционально: анимированные переходы между экранами.