# Level 0: The Weather

## Экосистема:

Требуется инициализировать базовое JS приложение (без использования каких-либо библиотек и фреймворков). Для этого приложения создать в том числе Webpack конфиг, который позволяет запускать это приложение с оптимизированным кодом. Webpack-конфиг сделать с минимизацией кода. То есть на выходе в билде должно получиться 2 файла: HTML и JS. Вне зависимости от того, сколько файлов было на этапе разработки.

Суть: реализовать **одностраничное** приложение с реализацией своего собственного роутера. Приложение должно включать в себя 2 страницы, ориентир дизайна - внизу на скриншоте. При этом приложение должно быть доступно в том числе и на планшетах, на десктопных устройствах (не только mobile view).

## 

## 

## Функциональные требования:

1. Приложение должно отображать информацию о погоде в выбранном городе или сохраненных городах.
2. При поиске города - вместо списка плашек текущих сохраненных городов, надо отображать простой список подсказок городов (к примеру, при поиске “New” нужно делать запрос к списку городов и выводить 3 самых подходящих по названию).
3. По нажатию на плашку сохраненного города - переходим на страницу этого города. Там показываем более подробную информацию, доступную с API. Должна быть возможность выйти на страницу всех городов.

## Технические требования:

1. Использование фреймворков не требуется. Писать надо на чистых JS-файлах. Использование JQuery тоже не разрешено. Файлы разбивать по соответствующим группам.
2. Должны быть обработаны все исключения, связанные с внешними интеграциями в приложении - пользователь должен знать о неудачных API запросах (можно просто выводить alert).
3. Переход между страницами должен быть мнимым - то есть надо реализовать свой аналог роутера - подписаться на состояние адресной строки, и скрывать / вставлять необходимый контент в зависимости от адресной строки.
4. Перед разработкой надо настроить ESLint для приложения. В качестве дополнения можно настроить Prettier и связать его с ESLint. *Будет дополнительным плюсом реализовать pre-commit hook, который не позволит сделать push в репозиторий, если в приложении присутствуют ошибки ESLint.*
5. Написание документации к проекту; в README-файле проекта перед сдачей задания должна быть написана краткая документация на английском или русском, состоящая из 4 пунктов: “Task” (ссылка на этот документ), “How to run the app” (инструкции по билду приложения), *плюсом будет написание краткого описания структуры папок: для каждой папки описать, какого типа файлы она хранит.*
6. Надо писать Unit-тесты используя Jest.

## Дополнительные баллы можно получить за:

1. Действительно удобный пользовательский интерфейс.
2. Реализованный Theme-менеджмент. Возможность с легкостью поменять цветовую схему приложения **из приложения** (надо реализовать переключатели для пользователя).
3. Вместо вывода alert - сделать вывод кастомных notification messages в правом верхнем углу экрана.