Whitelogic

Frontend Problem Statement

Deadline: As mentioned in the email

OVERVIEW

Build a mini weather app using the OpenWeatherMap API

INSTRUCTIONS

Try to complete all features. We will be very impressed if you implement feature **8**. If you are unable to complete all features, ensure that whichever features you implement work correctly and completely.

Do not worry if you cannot implement all features; we can discuss whatever you have done during our meeting.

For any help required, feel totally free to contact Samir.

RULES

1. **Must** only use the **OpenWeatherMap API**
2. **Do not** use any third party library for the weather API.
3. **Must** use the **Bootstrap UI Framework**
4. Code **must** be hosted on a **Github repository**
5. **If possible**, use the AngularJS framework for client logic (highly encouraged, but not mandatory)

SPECIFICATIONS

1. The app consists of a grid of 9 panels. Every panel shows the weather data of one city. Initially, all panels would be empty.

2. On clicking on an empty panel, a text box is shown on the panel itself, asking the user to enter the city name. On entering the name, the panel now shows weather data for that city.
3. Every panel should have an “Edit” button - on clicking this, the user should be able to change the city shown in that panel.
4. If there was an error in getting the data (eg. wrong city name), the panel itself shows the error and the user can re-enter the city name.
5. Every panel has a background picture that describes the weather at the city - eg. sunny/rainy/cloudy.
6. The panels should be in a grid and responsive - the panels should align themselves in a row with respect to the screen size: for small screens - less number of panels in a row and for large screens - more number of panels in a row.

BONUS SPECIFICATIONS

7. The weather data must be updated regularly - data displayed should be no more than 30 seconds old. This must **not** be done using a manual refresh button.
8. **Enable offline use** - on refreshing the page, **even when offline**, the page shows the last obtained weather data. To implement this, use either HTML5 LocalStorage or IndexedDB (use the Angular IndexedDB wrapper if using this).