********** ***Must Read*** **********

TestVagrant is a code-first organisation and we strive for high programming standards. We are looking for people who are passionate about solving technical problems with good programming ethics. Hence, we encourage you to build your framework keeping in mind the following factors:

1. Modularity
2. Readability
3. DRY principle(s) & DAMP tests
4. Scalability

These factors are the guiding principles for us to evaluate your submission in detail.
All the best!

****

Problem Statement - Weather Reporting Comparator

Create a test automation framework with following capabilities :

1. Can interact with web applications
2. Can interact with RESTful APIs
3. Can compare objects retrieved from UI & service layers

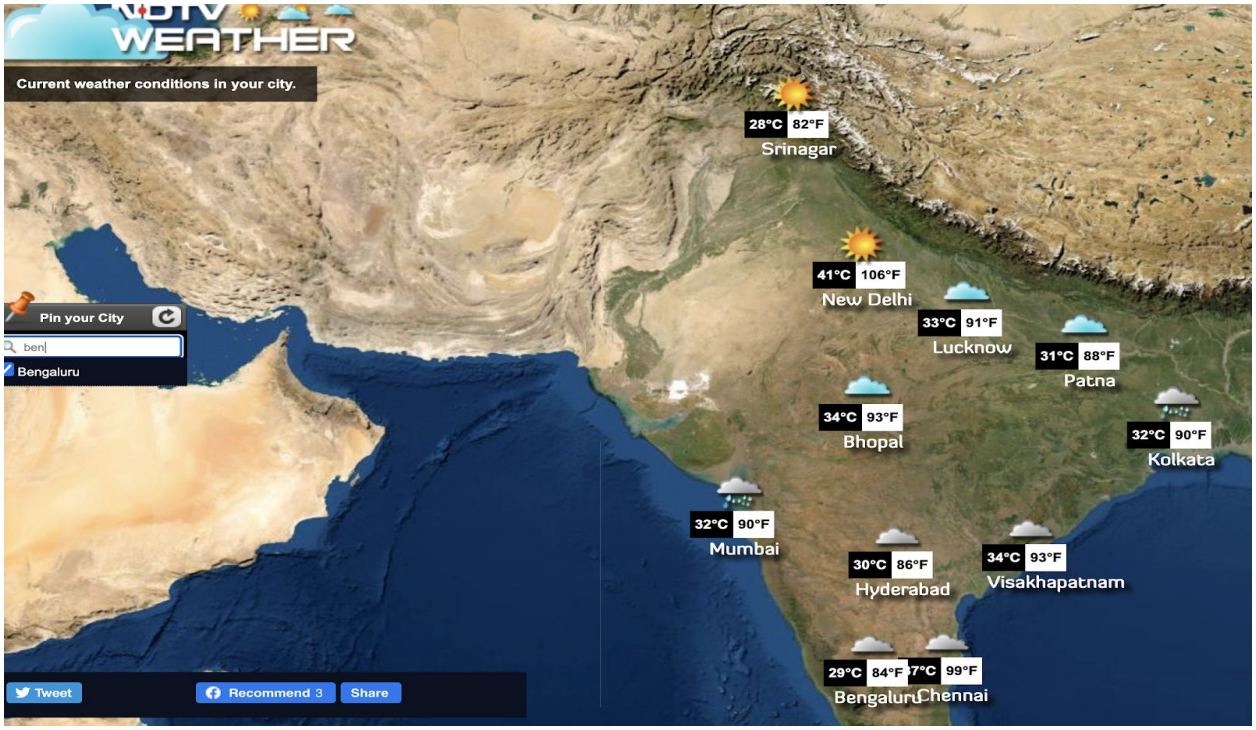
Focus area : The framework should compare weather information from different sources as under:

Source 1 : The website https://www.ndtv.com/.
Source 2 : The public weather API by https://openweathermap.org/

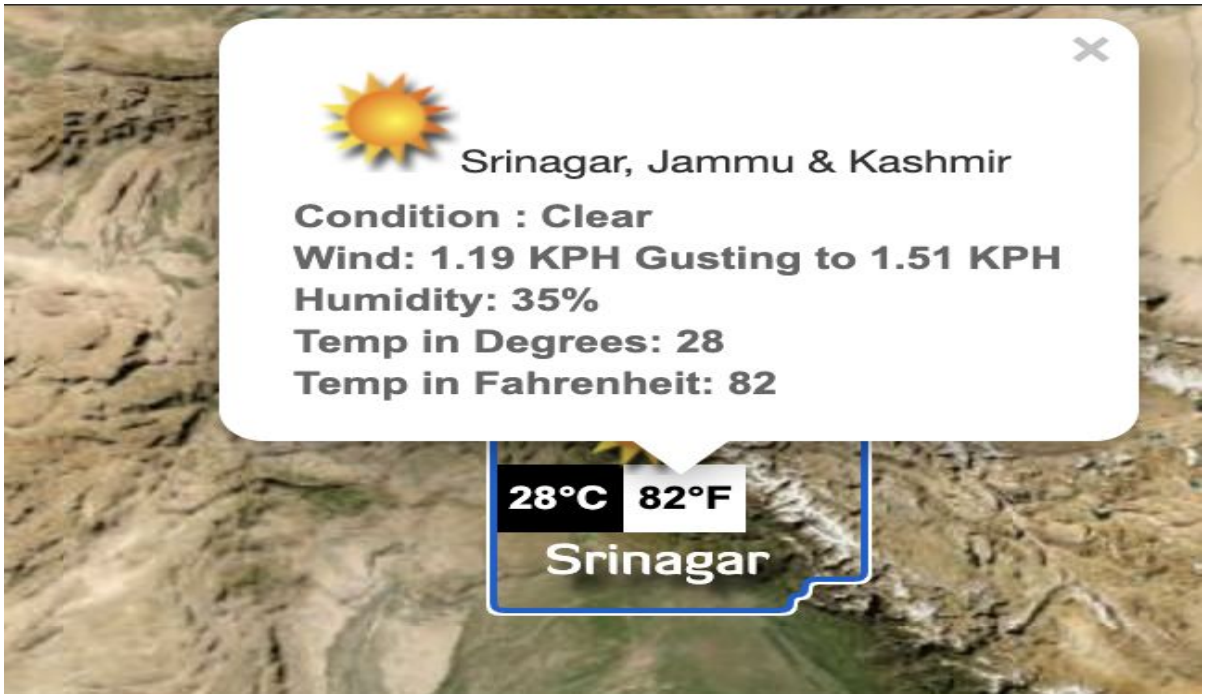
What to do?

Phase 1

1. Use any UI automation tool to automate https://www.ndtv.com/
2. Reach the weather section of the website



3. Use the `***Pin your city***` section on the left of the screen to search & select any given city
4. Validate that the corresponding city is available on the map with temperature information
5. Validate that selecting any city on the map reveals the weather details (sample screenshot below)



Phase 2

1. Use the APIs listed here (https://openweathermap.org/current) for getting current weather data for any city
   NOTE: Please use this value as API key in the request :
   "**7fe67bf08c80ded756e598d6f8fedaea**"

2. Build a scalable API handler - apart from processing the API to be used for fetching weather information, it should be able to accommodate variety of API requests(like endpoints, params, headers and HTTP methods) in future
3. Trigger the REST API (identified in step 1 above) to retrieve weather information using any Rest client

Phase 3

1. Create a **comparator** that matches the temperature information from the UI in phase 1 against the API response in phase 2 (ensure that comparison is done using same temperature unit)
2. Build a **variance** logic, which should be configurable using an external file, that returns a success if temperature difference is within a specified range, else return a matcher exception

Great to have(Expect some brownie points here) :

1. A regression test suite that covers good quality tests on both UI & API layers
2. Analyse other available weather conditions on both sources that can be compared and do the comparison following a similar variance logic.

Example Workflow(*Just for reference***)**

1. Visit ndtv website's weather page and search for Bangalore
2. Store weather object 1 w.r.t this Bangalore (e.g. temp value as 33 degree celsius)
3. Get response from the weather API for Bangalore
4. Store the API response and build the weather object 2
5. Specify the variance logic - for e.g. 2 degree celsius for temperature
6. Compare weather objects 1 and 2 along with the variance and mark tests as pass or fail based on comparator response

Submission Guidelines:

- Make sure the ***code is object-oriented*** and strictly adheres to the language conventions
- Share your code as a zip folder or upload the code to google drive/ dropbox and share the link with us. ***Make sure you have provided us with access***.
- Use any language and build tool you are comfortable with (Java, Javascript, C#, ruby,python etc.)
- ***Commit your code at regular intervals***, we will look for the sanctity of commit messages.
- Add a ***readme with clear deployment instructions*** for local execution