** *Must Read* ***

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

1. Modularity
2. Readability
3. DRY & DAMP principles
4. Scalability

Features of an Ideal Submission:

- Code is structured in logical packages
- DRY and DAMP practices are followed
- Tests do not have any hard coded data
- Execution of the tests can happen in parallel
- Solution should be CI ready

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

****

Problem Statement - Weather Reporting Comparator

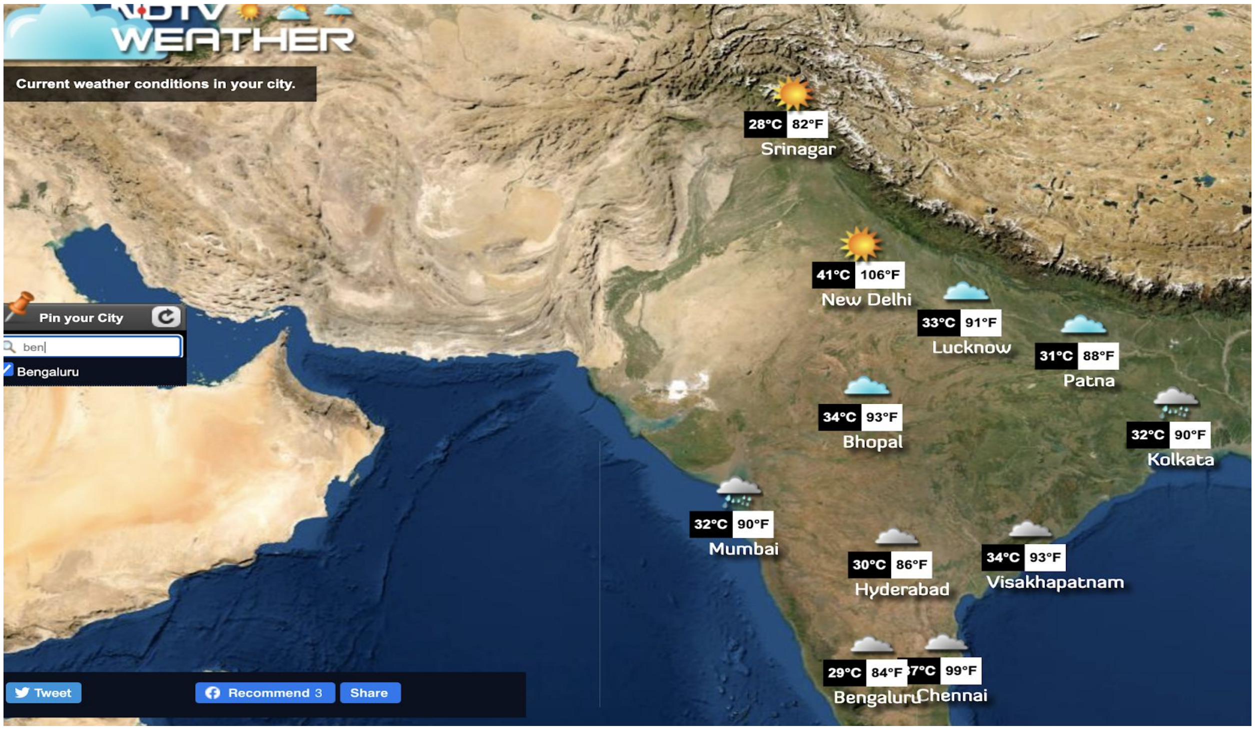
Create a test automation solution that compares weather information from web & API sources

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

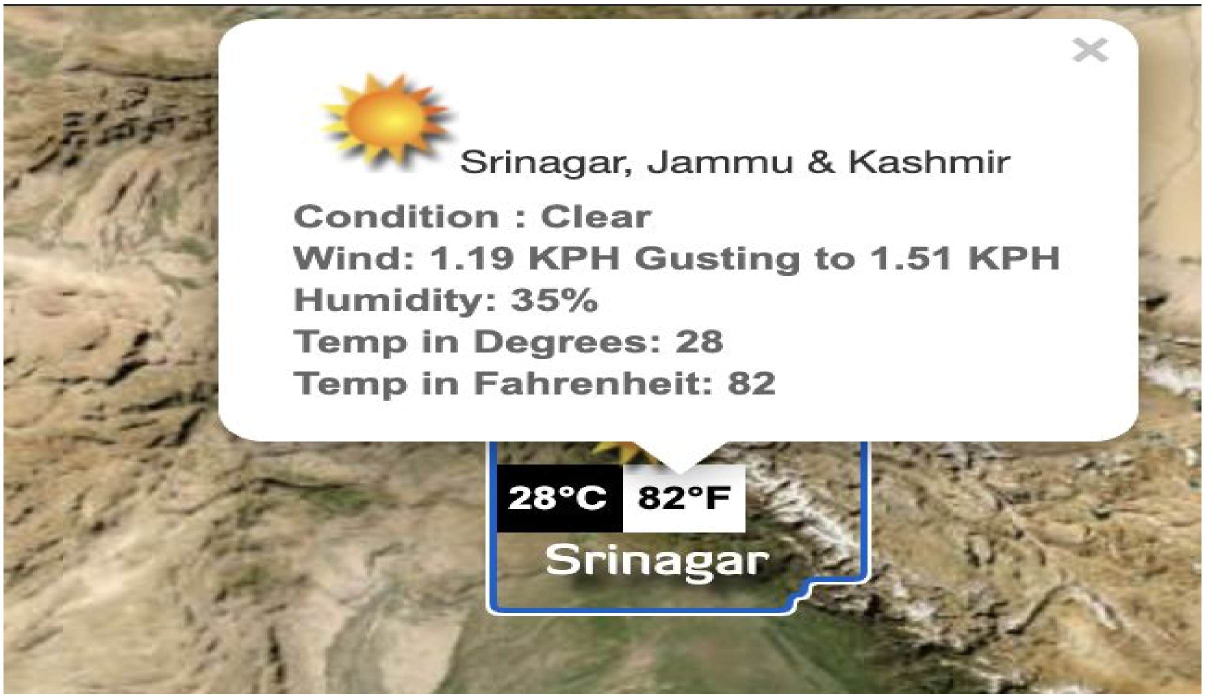
What to do?

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)



Srinagar, Jammu & Kashmir

Condition : Clear

Wind: 1.19 KPH Gusting to 1.51 KPH

Humidity: 35%

Temp in Degrees: 28

Temp in Fahrenheit: 82

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. Trigger the REST API (identified in step 1 above) to retrieve weather information using any Rest client

Phase 3

1. Implement a **comparator** logic 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. Use a configurable data file to hold magnitude of variation between UI & API response values and the test shall return a success if temperature difference is within a specified range, else return a user defined exception while marking the test as failed

Good to have(Expect some brownie points here) :

1. A regression test suite that covers good quality tests on both UI & API layers(independently)
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. Make necessary validations on the UI
3. Store weather object 1 w.r.t this Bangalore (e.g. temp value as 33 degree celsius)
4. Get response from the weather API for Bangalore
5. Make necessary validations on the API response
6. Store the API response and build the weather object 2
7. Specify the magnitude of allowed variation - for e.g. 2 for temperature - which means that temperature difference between API and UI layer should not be more than 2(in the the same unit of measurement)
8. 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