**Please find the below END TO END test case steps in Cucumber BDD Framework:**

Feature: Verifies first end-to-end test that all important information is present in homepage https://openweathermap.org/

Scenario: First End to End Test case to Verifies all important labels in homepage

Given I am on Open Weather map homepage

When User clicks on degree celsius toggle button

Then User should able to see degree celsius in page

Given User is at Open Weather map homepage

When User clicks on degree celsius toogle button and also find out the number of label in the entire page

Then User should get the list of total number of labels inthe entire page

Given User is at open Weather map homepage

When User clicks on labels to get display the complete name of all labels

Then User should able to fetch all labels names of the page

And user should able to validate the labels of entire page like "toggle candy blue" "Search" "highcharts-axis-labels" and so on

Feature: Verifies second end-to-end test that all important information for invalid city report is present in homepage https://openweathermap.org/

Scenario: Second End to End Test case to verify the invalid city searches in homepage

Given I am on Open Weather map homepage

When User enters an invalid city name "Invalidcityname" at search textfield

Then User should able to see the error message like "Not Found"

When user enters the city name "Somecityname" at Search textfield for the weather

Then user should able to see the weather report for Respective Somecityname

When User enters some invalid city name and checks the error message

Then User should able to see the city weather "Not Found" error message

Feature: Verifies Third end-to-end test that all important information for valid city report is present in homepage https://openweathermap.org/

Scenario: Third End to End Test case to verify the Valid city searches in homepage

Given I am on Open Weather map homepage

When User enters an Valid city name "Validcityname" at search textfield

Then User should able to see the error message like "Not Found"

When user enters the city name "Somecityname" at Search textfield for the weather

Then user should able to see the weather report for Respective Somecityname

When User enters some invalid city name and checks the error message

Then User should able to see the city weather "Not Found" error message

Feature: Verifies Third end-to-end test that all important information links is present in homepage https://openweathermap.org/

Scenario: Fourth End to End Test case to verify the links which is present in homepage

Given User is at open Weather map homepage

When User get number of links are present in homepage

Then User should able to fetch all links and its text with details of the page

When User clicks one by one links are present in homepage

Then User should able to see printing all the links and details

When user is clicks on links one by one to redirect the page and get title of the page

Then user should able to see the redirecting the link is opened and printing the title of the page

package practiceTestCases;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.firefox.FirefoxDriver;

public class FindAllLinksofopenweathermap {

public static void main(String[] args) {

WebDriver driver = new FirefoxDriver();

driver.get("<https://openweathermap.org/>");

java.util.List<WebElement> links = driver.findElements(By.tagName("a"));

System.out.println(links.size());

for (int i = 1; i<=links.size(); i=i+1)

{

System.out.println(links.get(i).getText());

}

}

}

**steps for Continuous Integration:**

1. **Version management**
2. **Decide a development strategy**
3. **Decide your delivery strategy**
4. **Decide an application version strategy**
5. **Build automation tool**
6. **Create and implement unit tests**
7. **Automatically build a test environment**
8. **Automatically test your most precious software parts functionally**
9. **Optionally create an acceptance environment**
10. **Optionally create an installation package**
11. **Create some error reporting**
12. **Backup your Build Automation**