Scenario:

This application displays the 5 day weather forecast for a given location.

**Features:**

* Enter city name, get 5 day weather forecast
* Select day, get 3 hourly forecast
* Select day again, hide 3 hourly forecast
* Daily forecast should summarise the 3 hour data:
  + Most dominant (or current) condition
  + Most dominant (or current) wind speed and direction
  + Aggregate rainfall
  + Minimum and maximum temperatures
* All values should be rounded down

Test case 1:

Step 1: Check if the user is able to enter the City Name.

Excepted result: User should be able to enter the City Name.

Step 2: Validate if the user is able to get 5 day weather forecast.

Excepted result: User should be able to get 5 day weather forecast.

Step 3: Check if the user is able to select the day.

Excepted result: User should be able to select the day.

Step 4: Check if the user is able to get 3 hourly forecast.

Excepted result: User should be able to get 3 hourly forecast.

Step 5: Check if the user is able to view the Daily forecast summarise and can validate 3 hour data.

Excepted result: User should be able to view the Daily forecast summarise and validate 3 hour data.

Step 6: Validate if the user is able to view the Most dominant (or current) condition.

Excepted result: User should be able to view the Most dominant (or current) condition.

Step 7: Validate if the user is able to view the most dominant (or current) wind speed and direction.

Excepted result: User should be able to view the most dominant (or current) wind speed and direction.

Step 8: Validate if the user is able to view the Aggregate rainfall.

Excepted result: User should be able to view the Aggregate rainfall.

Step 9: Validate if the user is able to view the Minimum and maximum temperatures.

Excepted result: User should be able to view the Minimum and maximum temperatures.

Step 10: Verify if the user is able to view the all values are rounded down.

Excepted result: User should be able to round down all the values.

Gherkin Keywords.

Scenario: This application displays the 5 day weather forecast for a given location.

Given: Application displays the 5 day weather forecast

When: Enter city name, get 5 day weather forecast

Then: Select day, get 3 hourly forecast

When: Select day again, hide 3 hourly forecast

Then: hide 3 hourly forecast

When: Daily forecast should summarise the 3 hour data

Then: Most dominant (or current) condition

Most dominant (or current) wind speed and direction

Aggregate rainfall

Minimum and maximum temperatures

When: All values should be rounded down

Then: Selected Values should be rounded down.

Test Script using Java Selenium Web driver TestNG Annotations.

Using the Chromedriver.

1 Step: Initialize the Web driver.

WebDriverManager.*chromedriver*().setup();

WebDriver driver = **new** ChromeDriver();

System.***out***.println("Webdriver Initiated");

driver.get("[http://openweathermap.org/forecast5](https://apc01.safelinks.protection.outlook.com/?url=http%3A%2F%2Fopenweathermap.org%2Fforecast5&data=02%7C01%7Cj.kumar4%40wipro.com%7Cabc71c6e89c2464b8add08d82d5432a2%7C258ac4e4146a411e9dc879a9e12fd6da%7C1%7C0%7C637309186934232055&sdata=wnySgtW2HMzl4iOXJgHr1W%2F6uGkHoLmjid8Wv2ukVDw%3D&reserved=0)");

Reporter.*log*("Browser Opened");

driver.manage().window().maximize();

assert.assert(“Actual title” , “Expected title”);

extent.attachReporter(htmlReporter);

ExtentTest test = extent.createTest("Openweathermap", "Valid login");

test.log(Status.***INFO***, "Browser is Opened");

driver.findElement(By.*cssSelector*("#search > input")).sendKeys("Chennai");

test.log(Status.***INFO***, "Enter the City");

driver.findElement(By.*cssSelector*("#menu > div.collapse.navbar-collapse.navbar-ex1-collapse > ul > li:nth-child(1) > a")).click();

test.log(Status.***INFO***, " Select day, get 3 hourly forecast");

driver.findElement(By.*cssSelector*("#menu > div.collapse.navbar-collapse.navbar-ex1-collapse > ul > li:nth-child(3) > a")).click();

test.log(Status.***INFO***, " Select day again, hide 3 hourly forecast");

Using the SELECT Class to validate the Dropdown fields.

WebElement dropdownfield = driver.findElement(By.*id*("card\_type"));

Select select = **new** Select(dropdownfield);

select.selectByVisibleText("Most dominant (or current) condition");

Thread.*sleep*(2000);

select.selectByVisibleText("Most dominant (or current) wind speed and direction");

Thread.*sleep*(2000);

select.selectByVisibleText("Aggregate rainfall");

Thread.*sleep*(2000);

select.selectByVisibleText("Minimum and maximum temperatures");

Thread.*sleep*(2000);

driver.findElement(By.xpath("/html/body/div/div/div[1]/div/div[1]/div/div/form/div/button")).click();

Thread.sleep(1000);

driver.close;

driver.quit;

extent.flush;

BDD scenarios should be clear and understandable and instead of having one long scenario to cover all test cases split them into multiple BDD scenarios.

Scenario: 1

* Enter city name, get 5 day weather forecast

Scenario 1 : Enter city name, get 5 day weather forecast

Given the url should be launched

And the chrome browser should be opened

And the Homepage should be displayed with all the valid details

When the The City name option is displayed

Then the City name should get 5 day weather forecast.

* Scenario 2 : Select day, get 3 hourly forecast

Given the url should be launched

And the chrome browser should be opened

And the Homepage should be displayed with all the valid details

When the Select day, get 3 hourly forecast

Then the 3 hourly forecast should be displayed.

* Scenario 3 : Select day again, hide 3 hourly forecast

Given the Url should be launched

And the chrome browser should be opened

And the Homepage should be displayed with all the valid details

When the user can hide 3 Hourly forecast

Then the 3 hourly forecast should be Hided.

* Scenario 4 : Daily forecast should summarise the 3 hour data

Given the Url should be launched

And the Chrome browser should be opened

And the Homepage should be displayed with all the valid details

When the user is able to Summarise the 3 hour data.

Then the 3 hour data should be displayed.

* + Scenario 5 : Most dominant (or current) condition

Given the Url should be launched

And the Chrome browser should be opened

And the Homepage should be displayed with all the valid details

When the user is able to view the most dominant condition.

Then the most dominant condition should be selected successfully.

* Scenario 6 : The Most Dominant speed and Direction should be selected.

Given the Url should be launched

And the Chrome browser should be opened

And the Homepage should be displayed with all the valid details

When the user is able to view the most dominant (or current) wind speed and direction

Then the most dominant (or current) wind speed and direction should be selected successfully.

* Scenario 7 : Aggregate rainfall should be selected.

Given the Url should be launched

And the Chrome browser should be opened

And the Homepage should be displayed with all the valid details

When the user is able to view the Aggregate rainfall

Then the Aggregate rainfall should be selected successfully.

* Scenario 8 : Daily forecast should summarise the 3 hour data

Given the Url should be launched

And the Chrome browser should be opened

And the Homepage should be displayed with all the valid details

When the user is able to view the Minimum and maximum temperatures

Then the Minimum and maximum temperatures should be selected successfully.

Scenario 9: All values should be rounded down should be selected.

Given the Url should be launched

And the Chrome browser should be opened

And the Homepage should be displayed with all the valid details

When the user is able to view the all values

Then the All values should be selected successfully.

Negative Scenarios:

* Scenario 10: The Most Dominant speed and Direction should be selected.

Given the Url should be launched

And the Chrome browser should be opened

And the Homepage should be displayed with all the valid details

When the user is not able to view the most dominant (or current) wind speed and direction

Then the most dominant (or current) wind speed and direction should not be selected successfully.

* Scenario 11 : Daily forecast should summarise the 3 hour data

Given the Url should be launched

And the Chrome browser should be opened

And the Homepage should be displayed with all the valid details

When the user is not able to summarise the 3 hour data.

Then the 3 hour data should not be displayed.

* Scenario 12 : Select day, get 3 hourly forecast

Given the url should be launched

And the chrome browser should be opened

And the Homepage should be displayed with all the valid details

When the Select day, get 3 hourly forecast is not displayed.

Then the 3 hourly forecast should not be displayed.

Using the Jenkins and Ant Build tool we can execute all the Test cases Triggered.



**Prerequisites for Environment Setup**

Following are the prerequisites required to set up with −

Java

Why we need − Java is a robust programming language. Cucumber supports Java platform for the execution.

How to install −

Step 1 − Download jdk and jre from the following link http://www.oracle.com/technetwork/java/javase/downloads/index.html

Step 2 − Accept license agreement.

Step 3 − Install JDK and JRE.

Step 4 − Set the environment variable as shown in the following screenshots.

Eclipse

Why we need − Eclipse is an Integrated Development Environment (IDE). It contains a base workspace and an extensible plug-in system for customizing the environment.

How to install −

Step 1 − Make sure JAVA is installed on your machine.

Step 2 − Download Eclipse from https://eclipse.org/downloads/

Step 3 − Unzip and Eclipse is installed.

Maven

Why we need − Maven is a build automation tool used primarily for Java projects. It provides a common platform to perform activities like generating source code, compiling code, packaging code to a jar, etc. Later if any of the software versions gets changed, Maven provides an easy way to modify the test project accordingly.

How to install −

Step 1 − Download Maven from the following link − https://maven.apache.org/download.cgi

Step 2 − Unzip the file and remember the location.

Step 3 − Create environment variable MAVEN\_HOME as shown in the following screenshot.

Step 4 − Edit Path variable and include Maven as shown in the following screenshot.

Step 5 − Download MAVEN plugin from Eclipse.

Step 6 − Open Eclipse.

Step 7 − Go to Help → Eclipse Marketplace → Search Maven → Maven Integration for Eclipse → INSTALL.

Configure Cucumber with Maven

Step 1 − Create a Maven project.

Go to File → New → Others → Maven → Maven Project → Next.

Provide group Id (group Id will identify your project uniquely across all projects).

Provide artifact Id (artifact Id is the name of the jar without version. You can choose any name, which is in lowercase). Click on Finish.

Process to Execute the Step definition file.

* Create a feature file.
* Create a step definition file.
* Create a JUnit runner to run the test.

Step definition maps the Test Case Steps in the feature files to code. It executes the steps on Application under Test and checks the outcomes against expected results. In order to execute step definition it must match the given component in a feature.