CUCUMBER (BEHAVIOUR DRIVEN DEVELOPMENT)

Important Instructions

• Go to https://vs.iiht.tech/

• Perform the below tasks assigned to you.

• Please note that the users are not allowed to copy anything from VM to system and

vice versa. You are requested to take the screen shot of each action you perform

and the output and add it to a word document and upload this word document on the portal in assignment section.

Cucumber Introduction

Cucumber is a tool that supports Behavior Driven Development (BDD). It offers a

way to write tests that anybody can understand, regardless of their technical

knowledge.

What is "Feature File"?

Features file contain high level description of the Test Scenario in simple language. It is known as Gherkin. Gherkin is a plain English text language

Feature File consist of following components -

• Feature: A feature would describe the current test script which has to be executed.

• Scenario: Scenario describes the steps and expected outcome for a particular test case.

• Scenario Outline: Same scenario can be executed for multiple sets of data using scenario outline. The data is provided by a tabular structure separated by (I I).

• Given: It specifies the context of the text to be executed. By using datatables "Given", step can also be parameterized.

• When: "When" specifies the test action that has to performed

• Then: The expected outcome of the test can be represented by "Then"

Example for feature file :

What is "Step Definition"?

Step definition maps the Test Case Steps in the feature files(introduced by Given/When/Then) to code. It which executes the steps on Application Under

Test and checks the outcomes against expected results. For a step definition to

be executed, it must match the given compoent in a feature. Step definition is

defined in java files under " /step\_definitions ".

Example for stepdefinition:

Gherkin Syntax

Gherkin is line-oriented language just like YAML and Python. Each line called

step and starts with keyword and end of the terminals with a stop. Tab or space

are used for the indentation.

In this script, a comment can be added anywhere you want, but it should start

with a # sign. It read each line a

fter removing Ghrekin's keywords as given, when, then, etc.

Example:

Feature: Login functionality of

xyz.com site.

Sce

Given: I am a xyz site user.

When: I enter username as username.

And I enter the password as the password

Then I should be redirected to the

Dashboard of xyz

Framework Architecture

First of all, you would be creating maven/ gradle/java project in eclipse

or IntelliJ IDE.If you would be creating the java project, then you referring the cucumber and selenium libraries by downloading and referring it through build path of your project. If you would working with Gradle/Maven Project, then you need to specify the cucumber and selenium dependencies under dependencies in pom.xml/build.gradle. Features Folder Once you have specified libraries in project then you would be creating a features folder in your project.Now you would be creating the flightSearch.feature under it for first assignment. Where in feature file , you would be defining the gherkin syntax

e.g.

Fe

ature: Flight Search on AirAsia

Scenario:

Searching one way flight

Given

When

Then

Similarly, you would be creating feature file for newtourtravel.feature

and automationpractice.feature under features for Assignment 3 and

Assignment 4 respectively.

Also,

for assignment 2 you would be defining the second scenario in

airasia.feature.

Stepdefinitions package

This

is the package name that you would be specifying or you can specify

any other packagename of your choice.

Under this you would be defining the clas

s for

stepdefinitions for each

feature file say AirasiaSteps.java,TravelSteps.java and

AutomationPracticeSteps.java,Hooks.java and Runner.java etc.

Hooks.Java

In this class,we would be defining the hooks which will execute before

and after of every scenar

io.

Basically, launching the browser and navigating to the particular site we

would specifying in the @Before hook.

Which will be executed before every scenario.

For our case we would be defining the tagged hoo

ks with tags specified in

notes of assignment.

There would be three @Before hooks we would be defining for our cases

by tags @airasia,@travel and @automationpractice. So that the

particular hook will get executed corresponding tagged scenario.

Also, we would be defining @After hook , where we would be

specifying

that driver.quit . so that our session of driver is closed after every

scenario.

Note : There is only single @After hook can work for all scenarios in our

case.

Example of Hooks.java

AirAsiaSteps.java

In this class, we would be defining

the mapping of feature steps of air

asia scenarios by using @Given , @When , @Then etc. as the step

specified in feature. Under this we would writing the logic to perform

particular step.

Example:

Similarly, we would be defining the mapping of other fe

ature steps in its

corresponding stepdefinitions file.

Junit/Testng Runner Class

(Runner.java)

This class will use the

Junit annotation

@RunWith(),

which

tells

JUnit

what is the

test runner class

. It more like a starting point for

Junit to start executing

your tests.

In this class you would be defining the cucumberOptions, which will help

you specify the features file or folder path, stepdefinitions path, Reports

format and location, tags to control the execution.

@CucumberOptions

Example for Runner.j

ava

Report

Once you will execute the runner.java then you would be able to see

execution reports in the format that you specified in plugin

(CucumberOption)

Example for

default Cucumber html report:

Example for

default Cucumber json report: