IJARCCE

ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 10, Issue 9, September 2021

DOI: 10.17148/IJARCCE.2021.10905

Automating Naukri Website in BDD Framework using Cucumber Tool

Poojitha Hegde[1], Arpitha Hegde[2]

MTech., Computer Network and Engineering, BMS College of Engineering, Bangalore, Karnataka, India[1]

MTech., Computer Network and Engineering, BMS College of Engineering, Bangalore, Karnataka, India[2]

Abstract: Testing plays an important role to assure quality of the product and to provide better business optimization. The following paper gives an overview regarding Behaviour Driven Development (BDD) framework using cucumber tool. The main purpose of this paper is to differentiate how BDD is more efficient than Test Driven Development (TDD).

Keywords*:* BDD, Cucumber, TDD, Testing.

I.INTRODUCTION

BDD betide from TDD, which allows collaboration between business team and technical team with less intervention in the software code. The main advantage of using this framework is it can be shared even to stakeholder. The various tools are available for creating and automating BDD user stories they are cucumber, JBehave, SpecFlow etc. But all the tools will be using Gherkin as a common language. In traditional approach, before sprint business analyst or product owner writes user stories and acceptance criteria and then commits the stories to the sprint. During the sprint, development, testing, automating, bug fixing and retesting is done. Whereas, in BDD approach, before sprint business analyst or product owner takes requirement and writes user stories and examples. Then 3 Amigos meeting is conducted to discuss about the user stories and examples. Then selected examples are documented in Gherkin language. During the sprint, development, automation, test execution, bug fixing and retesting is done.

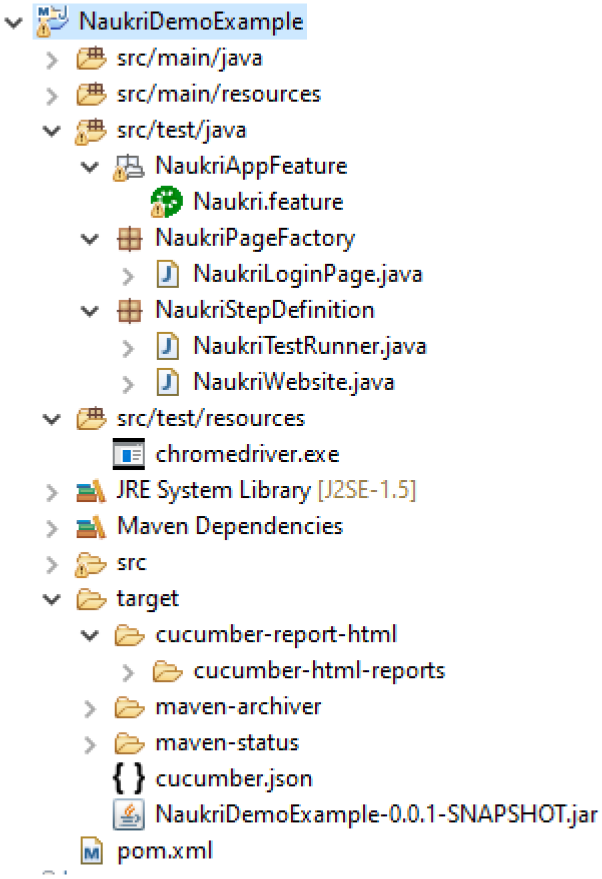
II.TECHNOLOGY USED

Eclipse: It is an open-source IDE (Integrated Development Environment) which is used to write code for the application.
Java: It is a programming language developed by sun microsystems in 1995.Around 3 billion devices will use java programming language like mobile device, e-business solution, games etc
Maven: It is a powerful tool, based on POM and it is used to build projects, dependency and documentation. The main purpose of this tool is in shortest period of time, the complete state of a development can be made.
Selenium: It is one of the most popular tools for automating functional aspects of web-based application, browser, technologies and platform.
Cucumber: It is one of the tools, that supports BDD framework. It supports various programming language like java, .Net, Ruby etc.

III.METHODOLOGY

In this paper, we automated Naukri website in BDD framework using Cucumber tool, by writing code in java programming language. Here we have created three package like NaukriAppFeature, NaukriPageFactory and NaukriStepDefinition. It consists of main file like Feature file, Step Definition file and Test Runner file to run Cucumber test scenario.

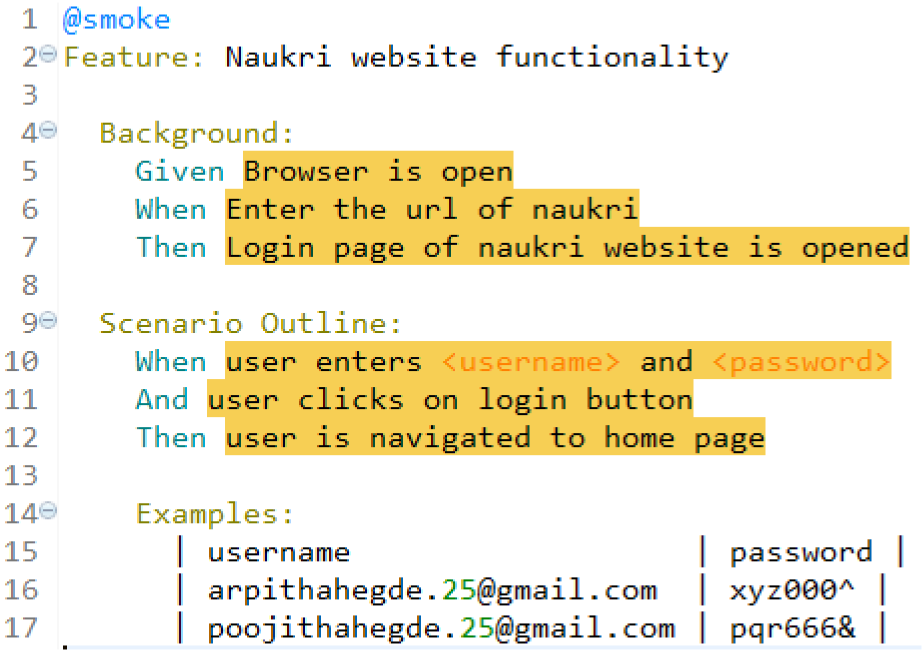
DOI: 10.17148/IJARCCE.2021.10905



```
NaukriDemoExample
  src/main/java
  src/main/resources
  src/test/java
    NaukriAppFeature
      Naukri.feature
    NaukriPageFactory
      NaukriLoginPage.java
    NaukriStepDefinition
      NaukriTestRunner.java
      NaukriWebsite.java
  src/test/resources
    chromedriver.exe
  JRE System Library [J2SE-1.5]
  Maven Dependencies
  src
  target
    cucumber-report-html
      cucumber-html-reports
    maven-archiver
    maven-status
    cucumber.json
    NaukriDemoExample-0.0.1-SNAPSHOT.jar
  pom.xml
```

Fig.1. Project Structure

The below figure,gives an overview of feature file which is written using Gherkin language.This language is an plain text,business readable language so this filecan be shared with non technical user.



```
@smoke
Feature: Naukri website functionality

  Background:
    Given Browser is open
    When Enter the url of naukri
    Then Login page of naukri website is opened

  Scenario Outline:
    When user enters <username> and <password>
    And user clicks on login button
    Then user is navigated to home page

    Examples:
      | username                  | password |
      | arpithahegde.25@gmail.com | xyz000^ |
      | poojithahegde.25@gmail.com | pqr666& |
```

Fig.2. Feature file

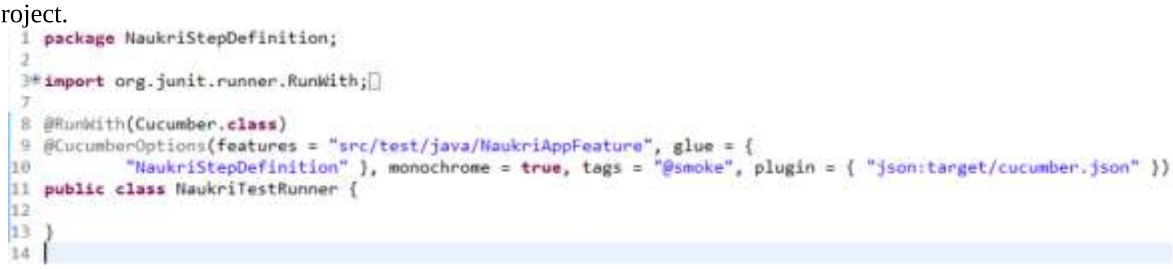
ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 10, Issue 9, September 2021

DOI: 10.17148/IJARCCE.2021.10905

The above file represents the Test Runner file.Where no code is written under this class but,it should include annotation like @RunWith and @CucumberOptions.This empty class provide a link between Feature file and Step Definition file of the project.

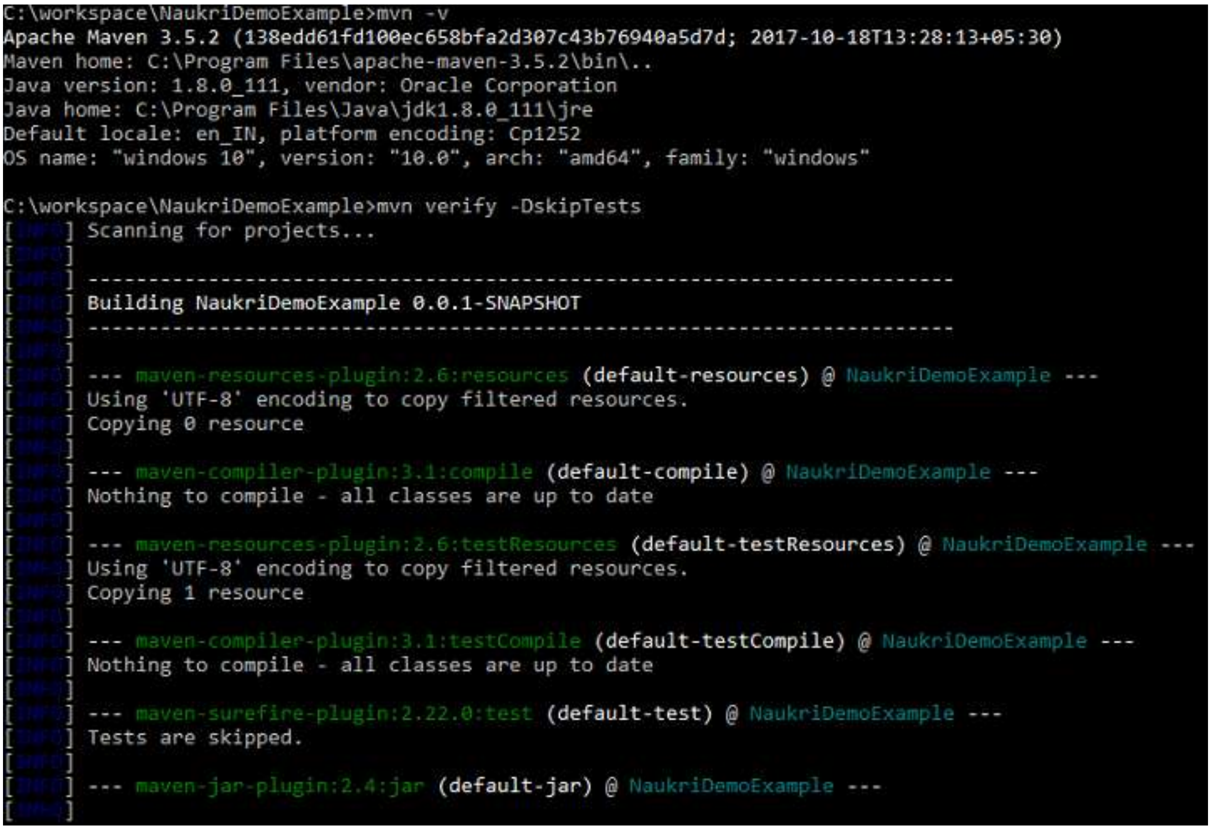


```
package NaukriStepDefinition;

import org.junit.runner.RunWith;

@RunWith(Cucumber.class)
@CucumberOptions(features = "src/test/java/NaukriAppFeature", glue = {
        "NaukriStepDefinition" }, monochrome = true, tags = "@smoke", plugin = { "json:target/cucumber.json" })
public class NaukriTestRunner {

}

```

Fig.3. Test Runner Class

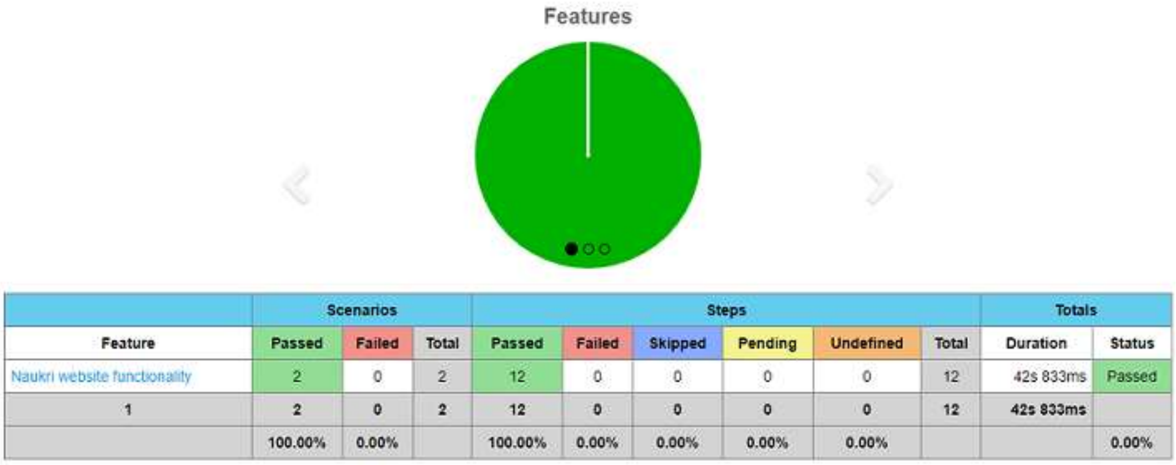
Page Factory is used to support page object design pattern using @FindBy annotation and initElements method.

IV.RESULTS



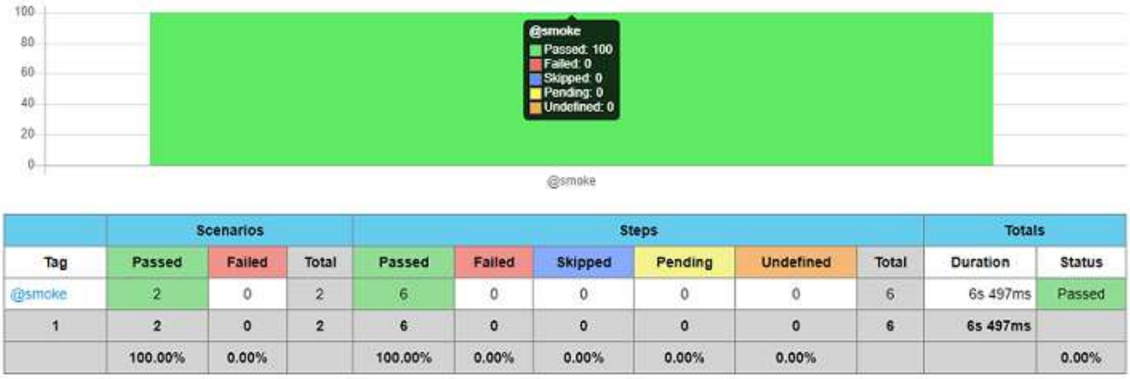
```
C:\workspace\NaukriDemoExample>mvn -v
Apache Maven 3.5.2 (138edd61fd100ec658bfa2d307c43b76940a5d7d; 2017-10-18T13:28:13+05:30)
Maven home: C:\Program Files\apache-maven-3.5.2\bin\..
Java version: 1.8.0_111, vendor: Oracle Corporation
Java home: C:\Program Files\Java\jdk1.8.0_111\jre
Default locale: en_IN, platform encoding: Cp1252
OS name: "windows 10", version: "10.0", arch: "amd64", family: "windows"

C:\workspace\NaukriDemoExample>mvn verify -DskipTests
[INFO] Scanning for projects...
[INFO]
[INFO] ------------------------------------------------------------------------
[INFO] Building NaukriDemoExample 0.0.1-SNAPSHOT
[INFO] ------------------------------------------------------------------------
[INFO]
[INFO] --- maven-resources-plugin:2.6:resources (default-resources) @ NaukriDemoExample ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] Copying 0 resource
[INFO]
[INFO] --- maven-compiler-plugin:3.1:compile (default-compile) @ NaukriDemoExample ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] --- maven-resources-plugin:2.6:testResources (default-testResources) @ NaukriDemoExample ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] Copying 1 resource
[INFO]
[INFO] --- maven-compiler-plugin:3.1:testCompile (default-testCompile) @ NaukriDemoExample ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] --- maven-surefire-plugin:2.22.0:test (default-test) @ NaukriDemoExample ---
[INFO] Tests are skipped.
[INFO]
[INFO] --- maven-jar-plugin:2.4:jar (default-jar) @ NaukriDemoExample ---
[INFO]
```

Fig.4. Maven Commands to Generate Report



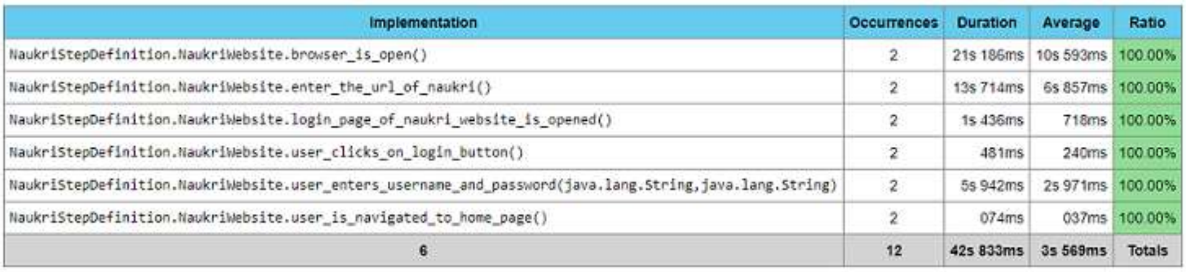
	Scenarios			Steps						Totals	
Feature	Passed	Failed	Total	Passed	Failed	Skipped	Pending	Undefined	Total	Duration	Status
Naukri website functionality	2	0	2	12	0	0	0	0	12	42s 833ms	Passed
1	2	0	2	12	0	0	0	0	12	42s 833ms	
	100.00%	0.00%		100.00%	0.00%	0.00%	0.00%	0.00%			0.00%

Fig.5. Feature Statistics



	Scenarios			Steps						Totals	
Tag	Passed	Failed	Total	Passed	Failed	Skipped	Pending	Undefined	Total	Duration	Status
@smoke	2	0	2	6	0	0	0	0	6	6s 497ms	Passed
1	2	0	2	6	0	0	0	0	6	6s 497ms	
	100.00%	0.00%		100.00%	0.00%	0.00%	0.00%	0.00%			0.00%

Fig.6.Tag Statistics



Implementation	Occurrences	Duration	Average	Ratio
NaukriStepDefinition.NaukriWebsite.browser_is_open()	2	21s 186ms	10s 593ms	100.00%
NaukriStepDefinition.NaukriWebsite.enter_the_url_of_naukri()	2	13s 714ms	6s 857ms	100.00%
NaukriStepDefinition.NaukriWebsite.login_page_of_naukri_website_is_opened()	2	1s 436ms	718ms	100.00%
NaukriStepDefinition.NaukriWebsite.user_clicks_on_login_button()	2	481ms	240ms	100.00%
NaukriStepDefinition.NaukriWebsite.user_enters_username_and_password(java.lang.String,java.lang.String)	2	5s 942ms	2s 971ms	100.00%
NaukriStepDefinition.NaukriWebsite.user_is_navigated_to_home_page()	2	074ms	037ms	100.00%
6	12	42s 833ms	3s 569ms	Totals

Fig.7. Step Statistics

V.CONCLUSION

In this paper we initially discussed about software testing and its types. Then we explore how BDD is more efficient than TDD. Finally, we discussed about the methodology of automating the application using cucumber and then results and reports are generated for the application.

REFERENCES

[1] H. Kaur and G. Gupta, “Comparative Study of Automated Testing Tools: Selenium, Quick,” Int. Journal of Engineering Research and Applications, vol. 3, no. 5, Sep-Oct 2013,.

[2] I. Singh and B. Tarika, “Comparative Analysis of Open Source Automated Software,” International Journal of Information & Computation Technology, vol. 4, Number 15 (2014).

[3] A. Bruns, A. Kornstadt and D. Wichmann, “Web Application Tests with Selenium,” IEEE, vol. 26, no. 5, 25 August 2009.

[4] S. . A. Sualim, N. . M. Yassin and R. Mohamad , “Comparative Evaluation of Automated User Acceptance Testing Tool for Web Based Application,” International Journal of Software Engineering and Technology.

[5] A. . A. and S. M. , “Web Application Testing: A Review on,” International Journal of Scientific & Engineering Research,, vol. 3, no. 2, February 2012.