AUTOMATION TESTING

(POINTMATTER SAAS)

An Internship Report submitted in partial fulfillment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

Submitted by

Meghana Botu, 1210316739

Under the esteemed guidance of

Mr. Rajesh Motapalukula

Project Manager, LogicMatter India Pvt Ltd

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

GITAM

(Deemed to be University)

VISAKHAPATNAM

JUNE,2019

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

GITAM INSTITUTE OF TECHNOLOGY

GITAM

(Deemed to be University)

DECLARATION

I, hereby declare that the internship review entitled "**TITLE OF THE INTERNSHIP**" is an original work done in the Department of Computer Science and Engineering, GITAM Institute of Technology, GITAM (Deemed to be University) submitted in partial fulfillment of the requirements for the award of the degree of B.Tech. in Computer Science and Engineering.

The work has not been submitted to any other college or University for the award of any degree or diploma.

Date:

Registration No.	Name	Signature
1210316739	Meghana Botu	

ACCEPTANCE LETTER FROM THE ORGANISATION/ COMPANY

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

GITAM INSTITUTE OF TECHNOLOGY

GITAM

(Deemed to be University)

CERTIFICATE

This is to certify that the internship report entitled "**PointMatter SaaS**" is a bonafide record of work carried out by **Meghana Botu (1210316739)** student submitted in partial fulfillment of requirement for the award of degree of Bachelors of Technology in Computer Science and Engineering.

SUPERVISOR **INTERNSHIP REVIEWER**

Mr. Rajesh Motapalukula

Project Manager

TABLE OF CONTENTS

ABSTRACT

Humans make mistakes all the time, especially when it comes to building a software. Some of those mistakes are unimportant, but some of them are expensive or dangerous. We need to check everything and anything we produce because things can always go wrong. Testing is required for an effective performance of software application or product. It's important to ensure that the application should not result in any failures because it can be very expensive in the future or in the later stages of the development.

Software Test automation makes use of specialized tools to control the execution of tests and compares the actual results against the expected result. Testing Tools not only help us to perform tests but also helps us to automate data set up generation, product installation, GUI interaction, defect logging, etc.

About Organization

LogicMatter Pvt. Ltd. is a Seattle based company, incorporated in 2011. It provides cloud-based information management, analytics, and applications services. It offers basic services to get going with Web-based and email delivered operational reports and dashboards, plus, services to move up and analyze with slice and dice, and drill down capabilities; and custom services to pick a set of services for particular situations.

LogicMatter has been helping IT teams and SIs/OEMs in Asia, Europe, and North America deliver value. They innovate to offer Cloud & IoT solutions to our partners and customers. They continue to invest in the latest Cloud, IoT, AI, Mobile, Visualization and Blockchain technologies and strive to build autonomous assist-systems to improve the lives of maintainers.

LogicMatter is the best place for freshers to work on each and every level of SDLC. The higher management is supportive to innovate new things and implement our ideas.
It's a good place to start and grow. They provide the employees and interns with a lot of opportunities to learn and improve themselves.

Schedule of Internship

The internship at LogicMatter Pvt. Ltd. was held from 2-May-2019 to 3-June-2019 (31 days).

Week-1 (2-May to 5-May): I was revised with Java concepts like polymorphism, inheritance, abstraction and encapsulation. I also installed JDK files and NetBeans.

Week-2 (6-May to 12-May): I was revised with software engineering and software testing, mainly manual and automatic testing. The whole intern was based on automatic software testing. In a span of week, I learned installation and working of TestLink Tool.

Week-3 (13-May to 19-May): I was taught about Selenium Testing. The selenium framework was used for automation in collaboration with TestLink Tool. After learning and trying a few experiments on code for login-logouts on well-known websites like Instagram, Facebook, I implemented the testing tool on our project PointMatter.

Week-4 (20-May to 26-May): Protractor Testing was taught, for which I revised JavaScript and converted the Selenium code to Protractor form i.e. from Java to JavaScript code. This tool is also integrated with TestLink.

Week-5 (27-May to 2-June): To automate the schedule of execution of code for PointMatter Portal, I learned Jenkins. I integrated Jenkins with Selenium which was previously integrated with TestLink. With the help of Jenkins with put up a schedule for execution which run automatically without our interference.

INTERNSHIP ACTIVITIES

TRAINING

During the course of the internship, we have received training on various software testing technologies and tools.
This included the theoretical concepts of "What is Software Testing?", "Manual Testing", "Automation Testing" and "Types of Automation Testing".
Further we were provided with training on Automation Testing Tools namely "TestLink", "Selenium", "Protractor", "Jasmine Reporter", "Jenkins".
Selenium and Protractor scripts were initially written and tested on Login Pages of 'Facebook', 'Twitter', 'Instagram', 'LinkedIn', 'Yahoo Mail', 'Times Job' and 'Naukri'.
Writing scripts for various types of websites gave us an exposure on how to write the automation testing scripts as well as connecting them to tools like Testlink and Jenkins to view the reports generated eased us while doing the same with the actual "PointMatter SaaS project".

ROLE IN APPLICATION DEVELOPMENT

My role during the summer internship at LogicMatter was 'Software Tester' for their current on-site project titled "PointMatter SaaS". During the period of 30 days, I tested various components of the project for bugs using Selenium Testing and Protractor while generating reports through TestLink as well as Jasmine Reporter and submitting the same to the higher authorities.
Although the internship was performed as a group of four, all four of us were given the same work as Software Tester as we are new to automation testing and everyone will gain enough practice on the concepts.

METHODOLOGIES AND FUNCTIONALITIES

1. Software Testing

Software testing is defined as an activity to check whether the actual results match the expected results and to ensure that the software system is defect free. It involves execution of a software component or system component to evaluate one or more properties of interest.

Software testing also helps to identify errors, gaps or missing requirements in contrary to the actual requirements. It can be either done manually or using automated tools. Some prefer saying Software testing as a White Box and Black Box Testing.

In simple terms, Software Testing means Verification of Application Under Test (AUT).

1.1 Automation Testing

Manual Testing is performed by a human sitting in front of a computer carefully executing the test steps.

Automation Testing means using an automation tool to execute your test case suite.

The automation software can also enter test data into the System Under Test, compare expected and actual results and generate detailed test reports. Test Automation demands considerable investments of money and resources.

Successive development cycles will require execution of same test suite repeatedly. Using a test automation tool, it's possible to record this test suite and re-play it as required. Once the test suite is automated, no human intervention is required. This improved ROI of Test Automation. The goal of Automation is to reduce the number of test cases to be run manually and not to eliminate Manual Testing altogether.

2. TESTLINK:

Test-link is most widely used web based open source test management tool. It synchronizes both requirements specification and test specification together. User can create test project and document test cases using this tool. With Test-Link you can create an account for multiple users and assign different user roles. Admin user can manage test cases assignment task.

2.1 INSTALLATION OF TESTLINK SOFTWARE:

Step1: Open https://bitnami.com/stack/testlink .

Step 2: Click on windows in the category of "On my computer".

Step 3: Click Download for Windows 64-bit.

Step 4: Sign-in your respective account or click on "No thanks, take me to the download".

Step 5: The download will start automatically.

2.2 CONFIGURATION OF TESTLIN

Step 1: Select installation language.

Step 2: Click Next takes you to Installation folder to install bitnami testlink stack.

Step 3: Create an admin account which is feasible to use and click next after you are done.

Step 4: If you want you can configure SMTP settings by setting mail support.

Step 5: After clicking Next, un-check the "Launch testlink in the cloud with Bitnami".

Step 6: Now click Install, the installation begins.

Step 7: After the installation check the "Launch bitnami testlink stack" and click finish.

Step 8: Bitnami page is loaded, click access testlink and enter the previously declared login credentials.

2.3 HOW TO CREATE A PROJECT IN BITNAMI

Step 1: Give a name, prefix and project description of the project you are creating. Then click on create

Step 2: Click on the testlink logo.

Step 3: Create Test plan by clicking on "Test plan management" and provide it with name and description.

Step 4: Now click on testlink logo (home page).

Step 5: Click on test specification which will create test suites and test cases.

Step 6: Click on settings button in test project.

Step 7: Click on create test suite which is denoted by "+" symbol.

Step 8: Create a test suite by providing the name and details and then click on save. Now the test suite is created.

Step 9: Click on the created test suite which is present on left panel of your window. The test suite is opened on the right panel.

Step 10: Now click on the settings button present on the top.

Step 11: To create test cases for the suite click on create denoted by "+" in test case operations.
Step 12: Provide the new test case with title, summary, preconditions, and click on create.
Step 13: Now the test case is created where you can click on "create step" and add steps required for the test case.
Step 14: Ensure that each test case step is provided with step action, expected results(optional) and click "save".
Step 15: After entering all the required steps click on "save and exit".

3. SELENIUM:

Selenium is a free (open source) automated testing suite for web applications across different browsers and platforms. It is quite similar to HP Quick Test Pro (QTP now UFT) only that Selenium focuses on automating web-based applications. Testing done using Selenium tool is usually referred as Selenium Testing.

It has four components.

Selenium Integrated Development Environment (IDE)

Selenium Remote Control (RC)

WebDriver

Selenium Grid

3.1 INSTALLATION OF SELENIUM:

Step1: Download and install JDK (java development kit), once the installation is complete open command prompt and type "java" then you should be able to see various classes and objects of java as the output.
Step 2: Install Eclipse IDE, during the installation click "Eclipse IDE for Java Developers", select a proper path for eclipse IDE and "launch" it.
Step 3: Download the "Selenium Java Client Driver" which is in the form of zip file so one has to unzip it you will get a jar file.
Step 4: Configure Eclipse IDE with webdriver after launching the eclipse, accept the default location for the work space.
Step 5: Then create a new java project. A pop-up will be opened where you need to enter the project name, location, layout project, execution JRE, click "Finish".

Step 6: Right click on the newly created project, Select New > Package
A pop-up window will open to name the package, Enter the name of the package and Click on Finish button.
Step 7: Create a new Java class under “newpackage” by right-clicking on it and then selecting- New > Class, and then name it as "MyClass".
Step 8: When you click on Class, a pop-up window will open, enter details as Name of the class, Click on Finish button.
Step 9: In this step, Right-click on "newproject" and select **Properties**. On the Properties dialog, click on "Java Build Path”, Click on the **Libraries** tab, and then, Click on "Add External JARs."
Step 10: When you click on "Add External JARs.." It will open a pop-up window. Select the JAR files you want to add. After selecting jar files, click on OK button. Select all files inside the lib folder then Select files outside lib folder also.
Step 11: Add all the JAR files inside and outside the "libs" folder then Finally, click OK , this concludes adding files to our project.

3.1.1 ADDING TESTNG:

TestNG:

TestNG is an automation testing framework in which NG stands for "Next Generation". TestNG is inspired from JUnit which uses the annotations (@).

- Using TestNG you can generate a proper report, and you can easily come to know how many test cases are passed, failed and skipped.

  Setting up a new TestNG Project

  Step 1: Click File > New > Java Project

  Step 2: Type "FirstTestNGProject" as the Project Name then click Next.

  Step 3: We will now start to import the TestNG Libraries onto our project. Click on the "Libraries" tab, and then "Add Library…"

Step 4: On the Add Library dialog, choose "TestNG" and click Next.

Step 5: Click Finish.

Step 6: To add them in your java project we just write “@Test” and from that drop down we can just add the TestNG file from it. (This has to be written where ever we want main function to begin)

3.2 INTEGRATION OF TESTLINK WITH SELENIUM:

Step 1: The TestLink software must be installed in the system.

Step 2: Next we have to enable the automation in testlink application

To enable the API key: Go to Project -> Open your project -> Select “Enable Test Automation” / “API Key”.

Step 3: On My settings -> Click on “Generate a new key” which will generate an API key we must be noted.

Step 4: Download and include TestLink API Jar file in your project build path library. The following files need to be included: testlink-api-client.zip, testlink-api-client-2.0.jar, xmlrpc-common-3.1.jar, xmlrpc-client-3.1.jar, ws-commons-util-1.0.2.jar.

Step 5: Add all these jar files to the java project which will be connected to the testlink project.

Step 6: Write your script as required correctly and import files if necessary.

Step 7: Execute Selenium test and update Test result in TestLink.

4. PROTRACTOR:

Protractor plays an important role in the Testing of AngularJS applications and works as a Solution integrator combining powerful technologies like Selenium, Jasmine, Web driver, etc. It is intended not only to test AngularJS application but also for writing automated regression tests for normal Web Applications as well.

4.1 PROTRACTOR INSTALLATION:

Before installation, we have to install selenium and NPM(Node.js). Selenium has been installed before or can be installed from the above procedure. We need to install node.js to install protractor.

4.1.1 Installation of node.js:

Step 1: Go to the site https://nodejs.org/en/download/ and download the necessary binary files.

Step 2: Click on the downloaded .msi file to start the installation. Click the Run button in the first screen to begin the installation.

Step 3: Click the "Next" button to continue with the installation.

Step 4: Accept the license agreement and click on the Next button.

Step 5: Choose the location where Node.js needs to be installed and then click on the Next button, enter the file location for the installation of Node.js. Click on the Next button to proceed ahead with the installation.

Step 6: Accept the default components and click on the next button.

Step 7: Next, click the Install button to start the installation, then click Finish.

4.1.2 Installation of Protractor:

Step 1: Open command prompt and type **"npm install –g protractor"** and hit Enter. This will download all the necessary files.

Step 2: Check the installation and version using **“Protractor --version”.**

Step 3: Update the Web driver manager. The web driver manager is used for running the tests against the angular web application in a specific browser.

webdriver-manager update

Step 4: Start the web driver manager. This step will run the web driver manager in the background and will listen to any tests which run via protractor.

webdriver-manager start

Now, if you go to the following URL (**http://localhost:4444/wd/hub/static/resource/hub.html**) in your browser, you will actually see the Web driver manager running in the background.

Now, after the installation the Protractor needs two files to run, a **spec** file and **configuration** file.

1. **Configuration file**: This File helps protractor to where the test files are placed (specs.js) and to talk with Selenium server (Selenium Address). Chrome is the default browser for Protractor.

2. **Spec file:** This File contains the logic and locators to interact with the application**.**

For execution, first we have to start webdriver manager, in command prompt. Then give command “ protractor conf.js”. This will give us an output having pass or fail prompt in the command prompt.

4.2 Generate reports using Jasmine Reporters:

Installation of Jasmine Reporters

Step 1: Execute the command, in command prompt.

npm install --save-dev jasmine-reporters@^2.0.0

Step 2: Check the installation folders in the directory**.** " Node_modules" should be available if it is successfully installed.

Step 3: the existing conf.js file should be changed to:

```
exports.config = {
    seleniumAddress: 'http://localhost:4444/wd/hub',
    capabilities: {
       'browserName': 'firefox'
    },
    specs: ['spec.js'],
   framework: 'jasmine2' ,
    onPrepare: function() {
       var jasmineReporters = require('C:/Users/RE041943/Desktop/guru/node_modules/jasmine-reporters');
       jasmine.getEnv().addReporter(new jasmineReporters.JUnitXmlReporter(null, true, true)
       );
   }
 };
```

Step 4: Open command prompt and enter command " protractor conf.js". junitresults.xml will be generated with the result as report form for the conf.js. Open the xml document and verify the result.

4.3 Generation HTML Report using Jasmine

Step 1: Create a new document package.json.

Step 2: Now, execute the following five command to install all the dependencies from the json file.

npm install protractor-jasmine2-screenshot-reporter --save-dev

npm install protractor-beautiful-reporter --save-dev

npm install jasmine-data-provider --save-dev

npm install moment --save-dev

npm install file-system --save

Step 3: Add the following code in the conf.js file.

```
//For time stamp name to html report.
var today = new Date(),
timeStamp = today.getMonth() + 1 + '-' + today.getDate() + '-' + today.getFullYear() + '-' +
today.getHours() + 'h-' + today.getMinutes() + 'm';

//For HTML Screen shots.
var HtmlScreenshotReporter = require('protractor-jasmine2-screenshot-reporter');

var reporter = new HtmlScreenshotReporter({
  dest: 'Target/Screenshots',
  screenshotsFolder: 'images',
  showSummary: true,
  cleanDestination: false,
  reportOnlyFailedSpecs: false,
  captureOnlyFailedSpecs: true,
  fileNameDateSuffix: true,
  userCss: 'my-report-styles.css',
  filename: 'TestCases-' + timeStamp + '.html'
});
```

The above code has to be added at the beginning of the file.

```
beforeLaunch: function () {
        return new Promise(function (resolve) {
        reporter.beforeLaunch(resolve);
```

```
        });
  },
```

The above code is added before onPrepare function.

```
afterLaunch: function (exitCode) {
        return new Promise(function (resolve) {
        reporter.afterLaunch(resolve.bind(this, exitCode));
        });
  }
```

The above code is added after onPrepare function.

In onPrepare function,

```
jasmine.getEnv().addReporter(reporter);
```

Has to be added or changed.

These are the changes to be made in conf.js .

Step 4: The following changes have to made in spec.js file

```
beforeAll(() => {
        jasmine.DEFAULT_TIMEOUT_INTERVAL = 1000000;
});
```

This function should be added at beginning in the describe method.

Step 5: Create a folder named target in project directory, where all the reports will be stored after execution.

Step 6: Create a folder screenshots and css in the target folder.

The screenshot folder will contain HTML files of reports.

5. JENKINS

Jenkins is an open source Continuous Integration server capable of orchestrating a chain of actions that help to achieve the Continuous Integration process (and not only) in an automated fashion.

Jenkins is entirely written in Java. Jenkins is a widely used application around the world that has

around 300k installations and growing day by day.

It is a server-based application and requires a web server like Apache Tomcat. The reason Jenkins became so popular is that of its monitoring of repeated tasks which arise during the development of a project. For example, if your team is developing a project, Jenkins will continuously test your project builds and show you the errors in early stages of your development.

5.1 Software Requirements:

- Since Jenkins runs on Java, you need either Java Development Kit (JDK) or Java Runtime Environment (JRE).

5.2 DOWNLOADING AND INSTALLING JENKINS:

Following steps should be followed so that to install Jenkins successfully:

Step 1: Got to https://jenkins.io/download/ and select the platform. In our case Windows

Step 2: Go to download location from local computer and unzip the downloaded package. Double-click on unzipped **jenkins.msi**

Step 3: In the setup screen, click Next.

Step 4: Choose the location where you want to have the Jenkins instance installed (default location is C:\Program Files (x86)\Jenkins), then click on **Next** button.

Step 5: Click on the Install button.

Step 6: Once install is complete, click Finish.

Step 7: During the installation process an info panel may pop-up to inform the user that for a complete setup, the system should be rebooted at the end of the current installation. Click on OK button when the Info panel is popping-up.

5.2.1 UNLOCKING JENKINS:

Step 1) After completing the Jenkins installation process, a browser tab will pop-up asking for the initial Administrator password. To access Jenkins, you need to go to browse the following path in your web browser.

http://localhost:8080

If you can access the above URL, then it confirms that Jenkins is successfully installed in your system.

Step 2) The initial Administrator password should be found under the Jenkins installation path (set at Step 4 in Jenkins Installation).

For default installation location to C:\Program Files (x86)\Jenkins, a file called **initialAdminPassword** can be found under C:\Program Files (x86)\Jenkins\secrets.

However, If a custom path for Jenkins installation was selected, then you should check that location for **initialAdminPassword** file.

Step 3) Open the file and copy the content of the **initialAdminPassword** file.

Step 4) Paste the password it into browser's pop-up tab (http://localhost:8080/login?form=%2F) and click on Continue button.

5.2.2 Customize Jenkins

You can also customize your Jenkins environment by below-given steps:

Step 1: Click on the "Install suggested plugins button" so Jenkins will retrieve and install the essential plugins

Jenkins will start to download and install all the necessary plugins needed to create new Jenkins Jobs.

Step 2: After all suggested plugins were installed, the "Create First Admin User" panel will show up. Fill all the fields with desired account details and hit the "**Save and Finish**" button.

Step 3: Once you have filled the above data, finally it will ask for URL information where you can configure the default instance path for Jenkins. Leave it as it is to avoid any confusions later. However, if another application is already using 8080 port, you can use another port for Jenkins and finally save the settings, and you are done with installation of Jenkins. Hit the "**Save and Continue**" button:

Congratulations! We have successfully installed a new Jenkins Server. Hit the "Start using Jenkins" button.

5.3 INTEGRATING SELENIUM TESTLINK CODE WITH JENKINS

5.3.1 Setting java path in jenkins

Step 1: Go to jenkins Dashboard

Step-2: Click on Manage Jenkins

Step-3: Click on Global Tool Configuration

Step-4: In the JDK section click on JDK Installations

Step-5: Enter JAVA_HOME in the name section and place the path of jdk from the computer in the JAVA_HOME section.

Step-6: Click on Save and Apply

5.3.2 Generating windows batch file

Step-1: Go to eclipse workspace where the selenium script integrated with testlink is written.

Step-2: Right click on the Java file in the project explorer and click on convert to TestNG

Step-3: A new window will open , Click Next.

This will create a testng.xml file in the project folder.

Step-4: Go to the project directory and place all the jar files which were imported earlier in a single folder called lib.

Step-5: Create a new text document and type in the following code

```
java -cp bin;lib/* org.testng.TestNG testng.xml
```

And save the file as “run.bat”

5.3.3 Integrating batch file with Jenkins

Step-1: Go to jenkins dashboard and click on new item

Step-2: Type the name of the project and click on Freestyle Project and click on OK button.

Step-3: A new Configure window will open , Click on Advanced options button

Step-4: Select use custom workspace checkbox and add the path of the project in the Directory section.

Step-5: Scroll down and go to Build Section, There in the Add Build Step drop box select Execute Windows Batch Command and enter run.bat in the command text field.

Step-6: Click on Apply and Save.

5.4 Executing Selenium Script From Jenkins:

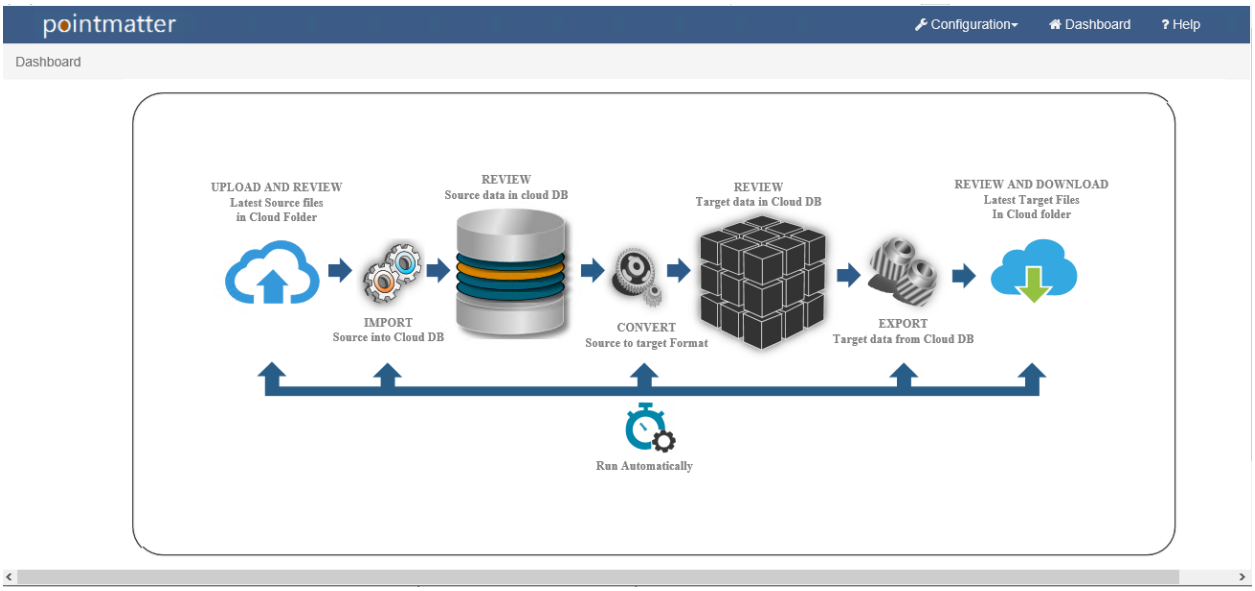
Step-1: Go to Jenkins Dashboard and select the project we want to execute.

Step-2: In the Project Dashboard click on Build Now button.

Step-3: The build will start and will be displayed in the build history section.

Step-4: Click on the date and time of the current build and it will open a new build dashboard , now select Console Output and the output will be displayed there.

6. Flowmatter Dashboard Testing



The Flowmatter Dashboard was tested for the proper functioning of buttons present on dashboard like Upload & Review, Import and Convert.

Selenium Script:

```
package pointmatter;

import org.openqa.selenium.WebDriver;
import org.openqa.selenium.By;
import org.openqa.selenium.JavascriptExecutor;
import org.testng.annotations.Test;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.support.ui.Select;

import testlink.api.java.client.TestLinkAPIClient;
import testlink.api.java.client.TestLinkAPIException;
import testlink.api.java.client.TestLinkAPIResults;
```

```
public class pmdash {

        public static String DEV_KEY = "0d900e8f3b0bbe34efcda7fe7c356b66";
        public static String SERVER_URL = " http://localhost/testlink/lib/api/xmlrpc/v1/xmlrpc.php";
        public static String PROJECT_NAME = "Pointmatter";
        public static String PLAN_NAME = "Pointmatter";
        public static String BUILD_NAME = "TestBuild";
        public static WebDriver driver;

        @Test
    public void TestOne() throws Exception{

        String result = "";
        String exception = null;

                System.setProperty("webdriver.chrome.driver","G:\\chromedriver.exe");

      String baseUrl = "https://i1.pointmatter.com:2424/t1_test/#/dashboard";
      String expectedTitle = "LogicMatter Inc";
      String actualTitle = "";

      try{
      driver = new ChromeDriver();
      driver.get(baseUrl);
      driver.manage().window().maximize();

      // get the actual value of the title
      actualTitle = driver.getTitle();
```

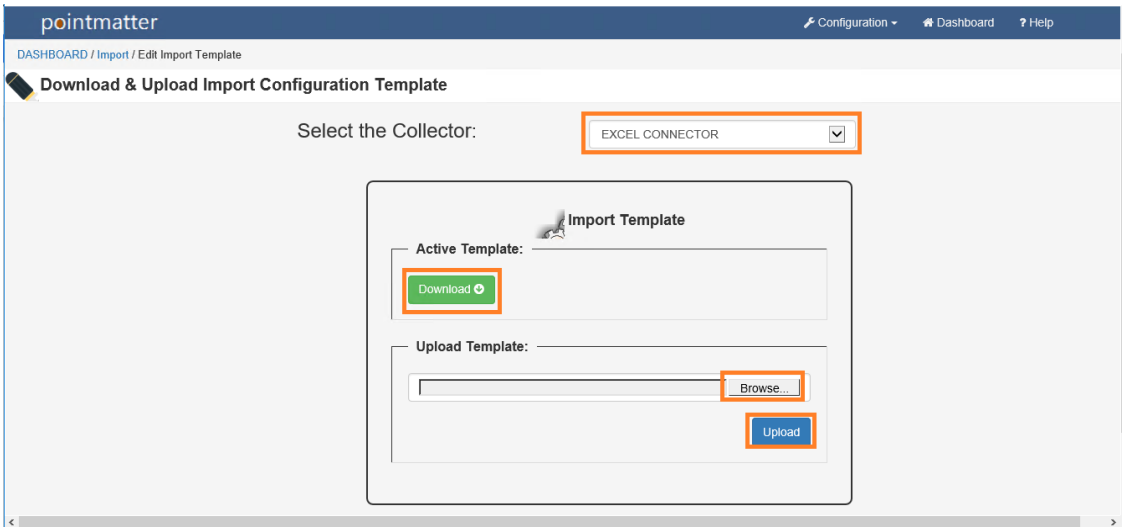
```
if (actualTitle.contentEquals(expectedTitle)){
   System.out.println("Title Passed!");
} else {
   System.out.println("Title Failed");
}
Thread.sleep(10000);
result = TestLinkAPIResults.TEST_PASSED;
updateTestLinkResult("Test-1", null, result);
}
catch (Exception e){
result = TestLinkAPIResults.TEST_FAILED;
exception = e.getMessage();
updateTestLinkResult("Test-1", exception, result);
}


try {
   driver.findElement(By.id("upload_sourcefiles")).click();
   driver.findElement(By.id("cnhContentPageBody_btnUplMdlShow")).click();
   Select drpSrc = new
Select(driver.findElement(By.name("ctl00$cnhContentPageBody$ddlSourceEntityFiles")));
   drpSrc.selectByVisibleText("T10Rep_T10Job_CriticalRoom_Location");

   driver.findElement(By.id("cnhContentPageBody_fluplSourceEntity")).sendKeys("G:\\intern\\fil
es\\Pasco_Critical_Room.xlsx");
   Thread.sleep(1000);
   driver.findElement(By.id("cnhContentPageBody_btnFileUpload")).click();
   Thread.sleep(3000);

   result = TestLinkAPIResults.TEST_PASSED;
   updateTestLinkResult("Test-2", null, result);
```

```
        }
        catch(Exception e) {
            result = TestLinkAPIResults.TEST_FAILED;
            exception = e.getMessage();
            updateTestLinkResult("Test-2", exception, result);
        }

        try {
            Thread.sleep(3000);
            driver.findElement(By.id("cnhContentPageBody_gvSourceFiles_lbFolderItem_0")).click();
            Thread.sleep(3000);

            result = TestLinkAPIResults.TEST_PASSED;
            updateTestLinkResult("Test-3", null, result);
        }
        catch(Exception e) {
            result = TestLinkAPIResults.TEST_FAILED;
            exception = e.getMessage();
            updateTestLinkResult("Test-3", exception, result);
        }

        try {
            driver.navigate().to("https://i1.pointmatter.com:2424/t1_test/Dashboard/ODSJobRunner.aspx");
            Thread.sleep(2000);
            driver.findElement(By.id("cnhContentPageBody_lnkbtnImpTem")).click();
            Thread.sleep(2000);
            Select drpCoc = new
Select(driver.findElement(By.name("ctl00$cnhContentPageBody$ddlImporter")));
            drpCoc.selectByVisibleText("EXCEL CONNECTOR");
            Thread.sleep(1000);
```

```
        driver.findElement(By.id("cnhContentPageBody_fluplImport")).sendKeys("G:\\intern\\files\\subscription_excel.xlsx");
        Thread.sleep(5000);
        JavascriptExecutor js = (JavascriptExecutor) driver;
        js.executeScript("window.scrollBy(0,1000)");
        driver.findElement(By.id("cnhContentPageBody_btnImportUpload")).click();
        Thread.sleep(2000);
        driver.switchTo().alert().accept();

        result = TestLinkAPIResults.TEST_PASSED;
        updateTestLinkResult("Test-4", null, result);
    }
    catch(Exception e) {
        System.out.println(e);
        result = TestLinkAPIResults.TEST_FAILED;
        exception = e.getMessage();
        updateTestLinkResult("Test-4", exception, result);
    }

    try {
        driver.navigate().to("https://i1.pointmatter.com:2424/t1_test/Dashboard/ADSJobRunner.aspx");
        driver.findElement(By.id("cnhContentPageBody_lnkbtnImpTem")).click();

        driver.findElement(By.id("cnhContentPageBody_fluplConvert")).sendKeys("G:\\intern\\files\\transformer_template.xlsx");
        driver.findElement(By.id("cnhContentPageBody_btnConvertUpload")).click();
        Thread.sleep(2000);
        driver.switchTo().alert().accept();

        result = TestLinkAPIResults.TEST_PASSED;
```

```
            updateTestLinkResult("Test-5", null, result);
        }
        catch(Exception e) {
            result = TestLinkAPIResults.TEST_FAILED;
            exception = e.getMessage();
            updateTestLinkResult("Test-5", exception, result);
        }
        driver.close();


            }


            private void updateTestLinkResult(String testCase, String exception, String result) throws
TestLinkAPIException{
            TestLinkAPIClient testlink = new TestLinkAPIClient(DEV_KEY, SERVER_URL);
            testlink.reportTestCaseResult(PROJECT_NAME, PLAN_NAME, testCase, BUILD_NAME,
exception, result);
            }
}
```

Screenshot:

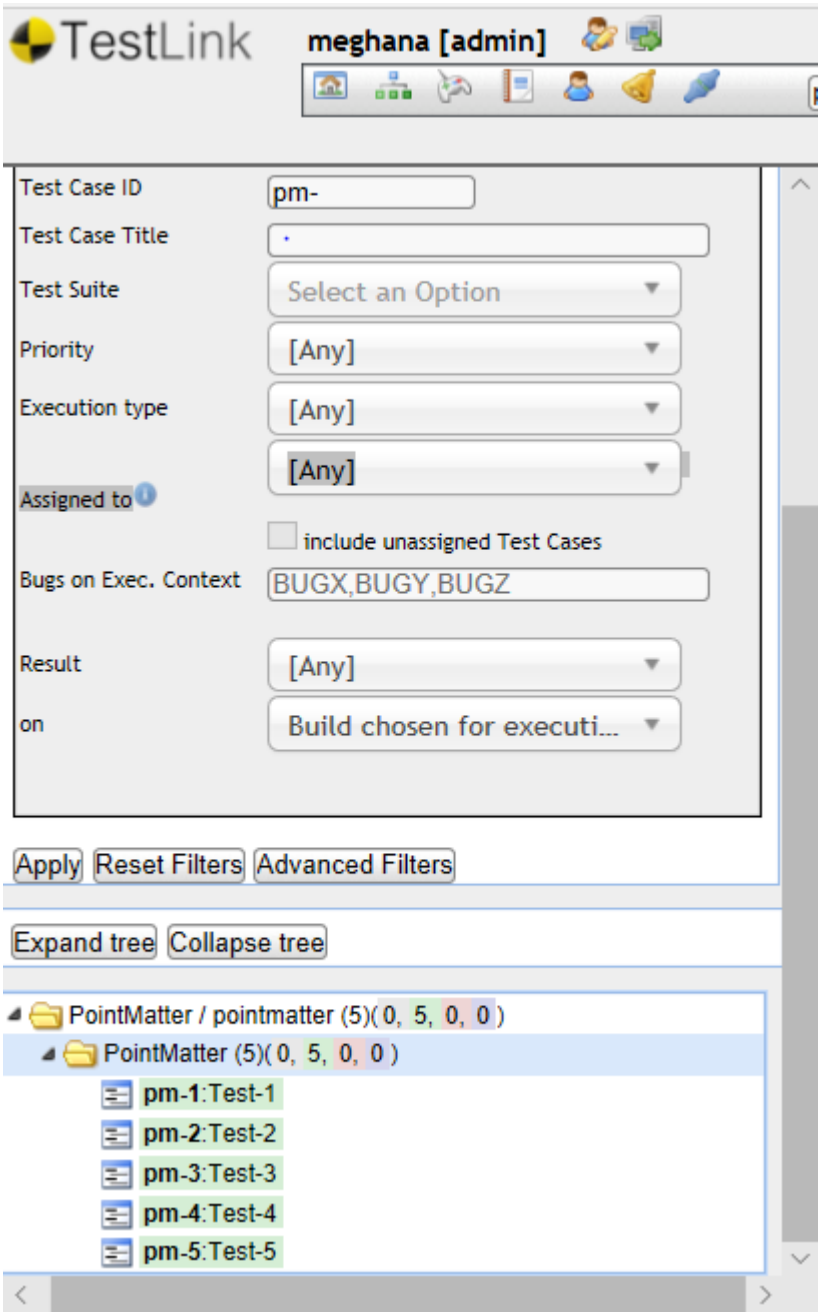


EXPERIMENTAL RESULTS:

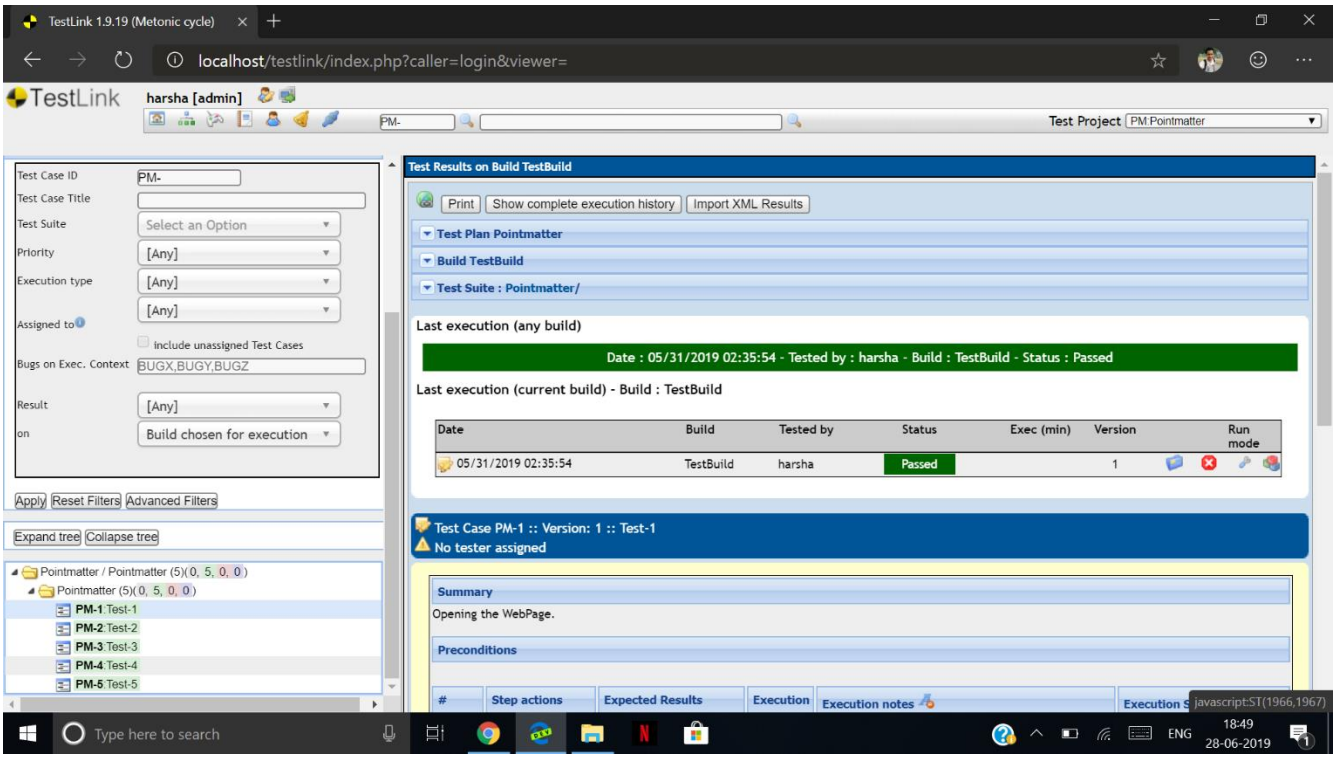
1)Execution using Selenium Integrated with test link:

We have learned and performed automated testing on “PointMatter” website where we have written the code in selenium and linked it with “TestLink” .

After execution the result looks like this in the test link:



The green highlighted part tells us that all our test cases have been executed.



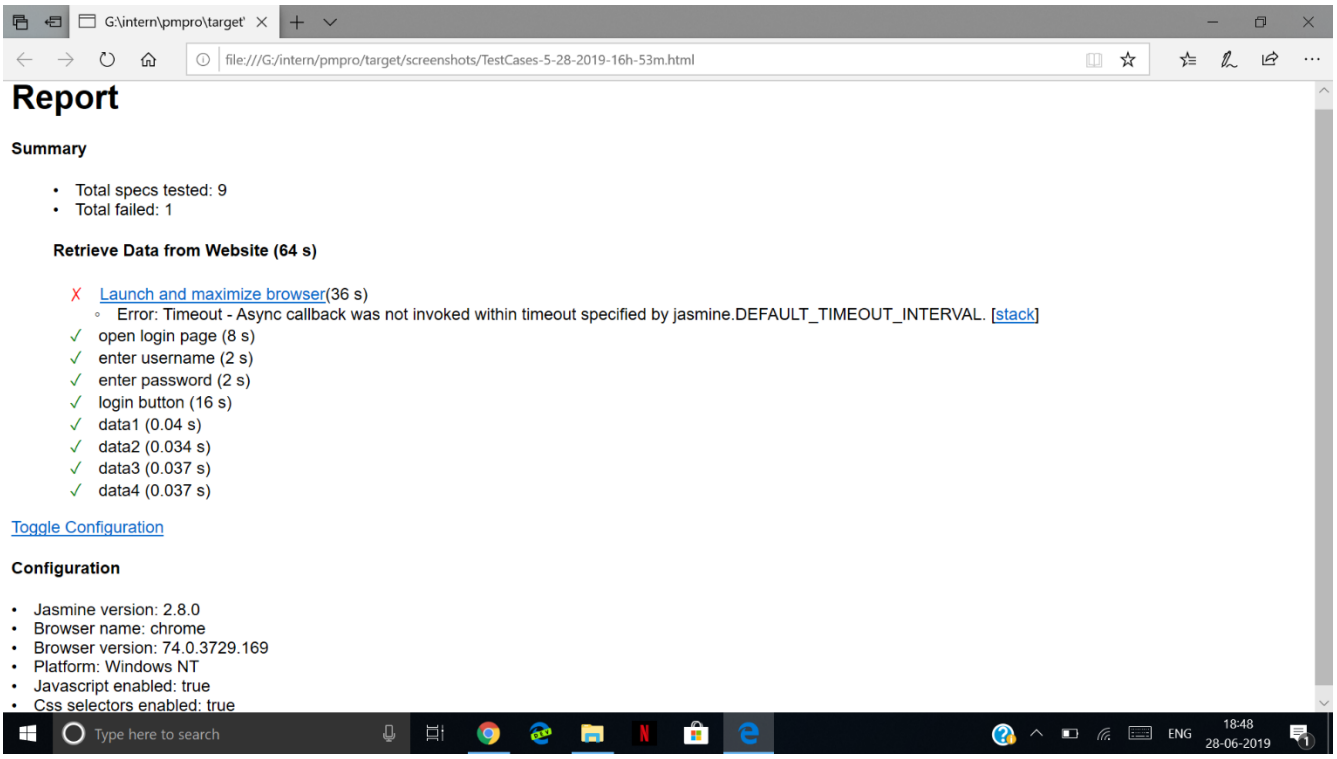
2) Execution using “Protractor” software:

This is another Automated Software where the code can be written on a document such as notepad,Visual Studio etc and kept in a folder. Using the folder’s path on our system we execute it on command prompt.

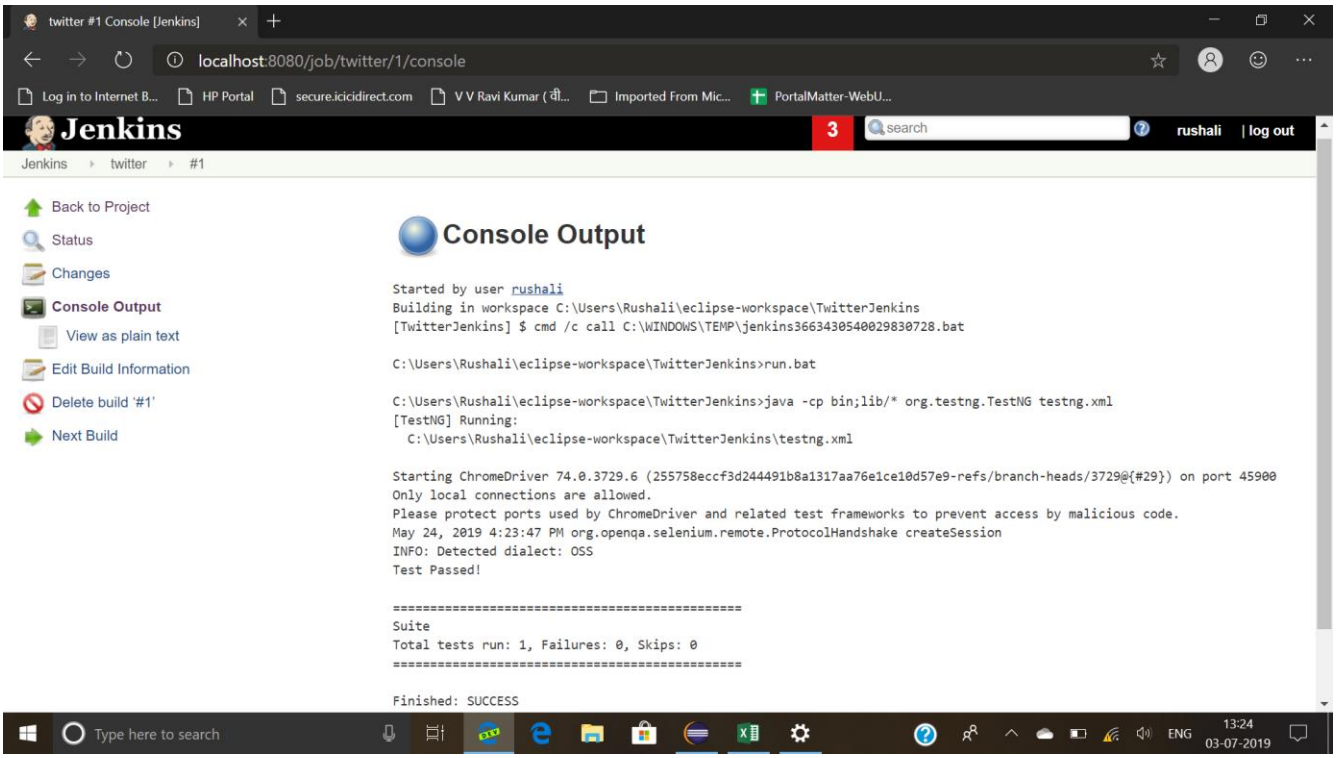
The successful execution looks like this:

```
C:\Users\user>G:

G:\>cd Intern

G:\Intern>cd firstprotractorfolder

G:\Intern\firstprotractorfolder>protractor conf.js
[19:16:55] I/launcher - Running 1 instances of WebDriver
[19:16:55] I/hosted - Using the selenium server at http://localhost:4444/wd/hub
Started
.....


5 specs, 0 failures
Finished in 24.737 seconds

[19:17:22] I/launcher - 0 instance(s) of WebDriver still running
[19:17:22] I/launcher - chrome #01 passed

G:\Intern\firstprotractorfolder>
```

The cursor points towards successful execution line.



3) Jenkins Report



INTERNSHIP CERTIFICATE

References:

A. Websites:

1. Step by Step Procedure for integrating Selenium with TestLink, https://www.corestack.io/blog/step-by-step-procedure-for-integrating-selenium-with-testlink/

2. Update TestLink Through Selenium, https://www.softwaretestinghelp.com/testlink-tutorial-3/

3. Integration of Selenium WebDriver with TestLink, https://blogs.perficient.com/2017/08/04/integration-of-selenium-webdriver-with-testlink/

4. How to Install TestNG framework, https://www.ecanarys.com/Blogs/ArticleID/169/How-to-Install-TestNG-framework-Step-by-Step-installation-process

5. TestNG Tutorial, https://www.guru99.com/all-about-testng-and-selenium.html

6. Protractor Testing Tutorial, https://www.guru99.com/protractor-testing.html

B. YouTube References:

7. Selenium Integration with Jenkins, https://www.youtube.com/watch?v=OxOaECjGxkg