**Miracle Application Development Assets**

**QA Automation Assets**

**For Internal Use ONLY**

**July 17, 2020**

**Version - 1.0**

**Srivarsha Marella**

**Manisha Aemula**

**Table of Contents**

[1.0 Objective 3](#_Toc45891927)

[2.0 Asset Details 3](#_Toc45891928)

[3.0 Tools & Reference Materials 3](#_Toc45891929)

[4.0 Asset Design and Architecture 3](#_Toc45891930)

[5.0 Asset Use 3](#_Toc45891931)

### Objective

This document outlines the test format used by Miracle Software Systems, Inc (Quality Assurance) group, and the test executions and results of our automation testing for Assets.

Initialization is the first step that must be undertaken before an application can be tested on a testing platform.

Our plan included performing testing as, UI Automation Testing Primary goal was to assess the web application functionality and permit test cases to be groomed where errors and omissions are evident.

The following is an overview of our test approach

QA (Quality Assurance) established a Statement of Record of QA Automation Assets in a spreadsheet.

In parallel automation test cases were identified and created.

Daily audits of the software configurations were supplied to the testers to ensure accuracy and consistency of the work environment.

We performed dry runs of the tests. Our goal was to ensure that the tests were initially performed and verified under controlled lab conditions, so that field resources were minimized when the tests were ultimately run in the field.

Finally, the QA team prepared the test plan results.

### Asset Details

1. **QA Automation Asset – Login/Sign in and Logout/Sign out Scenario**

|  |  |
| --- | --- |
| **Test Objective** | Login and Log out Scenario |
| **Author** | QA Automation Team |
| **Application** | <https://hubble.miraclesoft.com> |
| **Description** | Login to Miracle Software Systems internal application Hubble with respective credentials.  Validate functionality of email and password fields.  Able to click on login button.  Login to the application Successfully.  Navigate through the application, Logout of the application Successfully. |
| **Goal** | Able to login and logout of a web application using UI Automation script in Modular Driven Framework.  Reusable framework script for multiple login and logout functions across different platforms. |
| **Integration** | Jenkins with Selenium |
| **Global Run Status** | Passed |

1. **QA Automation Asset – Search Bar validation**

|  |  |
| --- | --- |
| **Test Objective** | Employee Search Validation |
| **Author** | QA Automation Team |
| **Application** | <https://hubble.miraclesoft.com> |
| **Description** | Post login scenario, navigate through the application to the search bar and find results using click function. |
| **Goal** | To be able to search for an employer (in this scenario) or an item (for e-commerce websites) and find the results of the search value by click function (.click). |
| **Integration** | Jenkins with Selenium |
| **Global Run Status** | Passed |

### Tools & Reference Materials

Steps to setup the Tech stack (Maven + Selenium + TestNg + Jenkins)

|  |  |  |  |
| --- | --- | --- | --- |
| S. No | Steps | Website (URLs) | Description |
| 1 | Install Java | <https://java.com/en/download/> | Used Java1.8 For this Project |
| 2 | Set the environmental | <https://www.java.com/en/download/help/path.xml> | “path” variable to ${Path\_to\_Java\_folder}/java/bin |
| 3 | Install NodeJS | <https://nodejs.org/en/download/> | Comes with npm package which is used for different tools as a prerequisite |
| 4 | Install Eclipse | <https://www.eclipse.org/downloads/> |  |
| 5 | Install Apache Maven – download binary zip folder | <https://maven.apache.org/download.cgi> | Download the binary and place it in “C drive” with admin privileges |
| 6 | Install TestNg plugin in the Eclipse | <https://dl.bintray.com/testng-team/testng-eclipse-release/> | Install from update website  * Select *Help / Install New Software...* * Enter the update site URL in "Work with:" field: * Update site for release: <https://dl.bintray.com/testng-team/testng-eclipse-release/>. * Make sure the check box next to URL is checked and click *Next*. * Eclipse will then guide you through the process. |
| 7 | Install maven builds in eclipse | <http://download.eclipse.org/technology/m2e/releases> | Install from update website  * Select *Help / Install New Software...* * Enter the update site URL in "Work with:" field: * Update site for release: [<http://download.eclipse.org/technology/m2e/releases>/](https://dl.bintray.com/testng-team/testng-eclipse-release/). * Make sure the check box next to URL is checked and click *Next*. * Eclipse will then guide you through the process. |
| 8 | Download Jenkins | <https://www.jenkins.io/download/> | Download Generic Java Package (.war) file |
| 9 | Executing Jenkins server |  | Navigate to the Jenkins (.war) file directoryRun “java -jar ${absolute\_path\_to\_the\_.war\_file}Wait for the console output “Jenkins is up and running successfully” |

|  |  |  |  |
| --- | --- | --- | --- |
| 10 | Enter the first-time login set up password |  | For Apple (Mac) users type in below command in Terminal to display the password: cat /Users/$(username)/.jenkins/secrets/initialAdminPassword |
| 11 | Create the first profile in the Jenkins dashboard |  | By default Jenkins runs on port 8080, Hence open <http://localhost:8080> in the browser to open the Jenkins DashboardCreate a profile and click on save and finish |
| 12 | Click on Manage Jenkins tab on the left in the dashboard |  |  |
| 13 | Install Maven Plugins - [Maven Integration plugin](https://plugins.jenkins.io/maven-plugin) |  | Select “Maven Integration plugin” and click on Install without restartWait to complete installation. |
| 14 | Click on Global Tool configuration |  |  |
| 15 | Set JAVA\_HOME path in JDK section |  | Click on JDK installationUncheck Install automatically optionSet name as “JAVA\_HOME” and value ${Java\_installation\_directory}/Java/Jdk\_versionFor example in windows: C:\Program Files\Java\jdk1.8.0\_241And click on APPLY and then SAVE |
| 16 | Set MAVEN\_HOME path in “Maven” section |  | Click on Maven InstallationsUncheck the “Install automatically”Add name MAVEN\_HOMEGet the Apache maven folder path we downloaded in step 5MAVEN\_HOME value ${Path\_to\_Apache\_Maven folder}Click on APPLY and SAVEFor example : In windows |
| 17 | Restart the Jenkins server |  | Shutdown the server using URL: <http://localhost:8080/exit>If displays “This URL requires POST”, then click on Retry using POST option. This will shutdown the server.Restart the server by typing in command line java -jar ${absolute\_path\_to\_Jenkins\_war\_file}Wait for the console output “Jenkins is up and running successfully”By default Jenkins runs on port 8080, Hence open <http://localhost:8080> in the browser to launch the Jenkins Dashboard |
| 18 | Create a new Eclipse Maven project |  | Click on file->New->ProjectSelect Maven->Maven ProjectSelect create a simple project without archetypeEnter grouId, Artifact Id and Name, Description of the project |

### Asset Design and Architecture

Framework – Modular driven automation framework

|  |  |
| --- | --- |
| **Create POM (page object model) reusable class for the respective test suite** | Granular level functions – Finding web elements, Sending input to the web elements, interaction with the web driver manager.  Calls functions from application.properties |
| **Create Modular reusable class - Scenarios** | Includes Scenario level functions which has sequence of steps certain functionalities (login and logout).  Calls granular functions from POM class.  Calls functions from application.properties |
| **Create Main class** | Calls scenario functions from modular class. |
| **application.properties** | All Modifiable variable/constant values such as web elements Xpath, CSS selectors, URLs values |
| **Pom.xml** | Automatically generated – Adding dependencies such as web driver manager, TestNg, Selenium Chrome Driver. |

### Asset Use

1. **QA Automation Asset – Login/Sign in and Logout/Sign out Scenario**

Validations - Email/Username, Password, Login, Logout/Sign out

Able to login and logout of a web application using UI Automation script in Modular Driven Framework. Reusable framework script for multiple login and logout functions across different platforms.

1. **QA Automation Asset – Search Bar validation**

Validations – Search Bar, Type in the value, Search Button (Click function)

Post login scenario navigate through the application to the search bar and find results using click function.

To be able to search for an employer (in this scenario) or an item (for e-commerce websites) and find the results of the search value by click function (.click).