MIRO WEBSITE AUTOMATION TEST REPORT

VERSION 1.0

***Introduction:***

The purpose of this document is to give high level overview about testing roadmap that will be followed for testing of **Miro website(Test Assignment)**. This document will list the detailed understanding about the Testing workflow that will be followed during Testing Life cycle.

# Scope of Testing:

Testing is based on the scenarios provided in the attached document.



# Test Strategy:

This section lists down the strategy to complete the testing the form which allows a user with a “Free” account to upgrade to “Team” .

1. Manually explore application to understand basic functionality.
2. Identify features in application.
3. Design test scenarios for testing of above mentioned form
4. Identify automatable test scenarios and mark the test cases accordingly.
5. Design test scripts for automation testing
6. Execute both automation tests against application in test.
7. Analyze test results to identify defects in application.

# Tools used for automation:

List of all the software and tools required for developing and executing automation framework.

1. Java (version 1.8)
2. Selenium (version 3.6.0)
3. TestNG (version 6.11)
4. Maven (version 4.0.0)
5. Log4j (version 1.2.17)
6. Google Chrome (version 79.0.3945.130)

# Automation Scripts Execution Steps:

To execute test scripts please follow below steps:

1. Install Java & maven and set their respective paths in system variables.
2. Extract the zip file
3. Open COMMAND PROMPT (cmd)
4. Navigate to the folder ‘MiroApp’
5. Clean and compile project using

*mvn clean compile*

1. To execute automation scripts, execute any of the following commands from command prompt:
   1. To execute all automation scripts (by default it will run on Chrome) mvn test
2. Logs available in Log folder
3. Test reports are available in below location

# \target\test-output\emailable-report

# \target\test-output\index.html

# Test Deliverables:

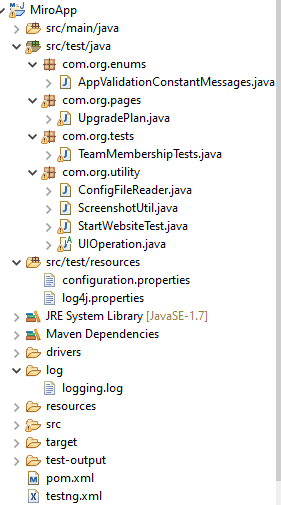
1. ***Mapping File*** *: Test Cases to Test script mapping file along with its status.*
2. ***Automated Details:*** *Automation test scripts are provided in the folder.*

# Automation Project structure:

This project is implemented in Page Object Model (POM) which helps in making the code more readable, maintainable, and reusable. Mentioned below are the description about different components of framework:

1. ***pom.xml:*** This file is used to perform build actions(clean, compile, install, test etc.) on project. All dependencies are included in pom.xml file.
2. ***drivers:*** This folder contains browser driver executables for chrome
3. ***log:*** This folder contains auto-generated log file.
4. ***com.org.enums:*** This folder contains common constants and enums which are present throughout the application.
5. ***Com.org.pages:*** This folder contains files corresponding to each functionality and has locators and methods to be used in the test methods.
6. ***Com.org.tests:*** This folder contains test files corresponding to each functionality. Each file contains tests to test the functionality end to end.
7. ***Com.org.utility:*** This folder contains classes to be utilized throughout the framework and is responsible to instantiate driver, read from properties file etc.
8. ***resources:*** This folder contains properties files which has details about the test application(url, username, password) and log4j properties.

*Project Structure:*



# TestNG Result:

# 1. Verify emailable-report at below location

# test-output\emailable-report

# 

# 2. Verify testing reports here

# test-output\index.html

# 