Home Loan Calculator Application

Automation Test Plan

I. Introduction

Upon the recommendation of the Senior Vice President and Chief Technical Officer of **KeefeRance Leasing and Finance Corporation**, the management approved the enhancement to the existing web page of the company to virtually assist potential home loan applicants in computing their monthly amortization if they desire to engage with the company in securing a short to medium term home loan assistance. This enhancement intends to assist the company in efficiently driving more home loan requests by reducing call center operation assistance cost 24/7 and in turn securing answers directly thru the page, 24/7. This document shall supplement the **Functional Specifications** document approved by the management and which shall form the basis for validating the quality of the Home Loan Calculator Application.

II. Scope

The scope of this document is to provide the test plan for the end to end UI and Functional Testing and is meant to be run autonomously and automatically thru scheduled execution time or as triggered by recent code commits from the development side on the home calculator application. Screenshots and videos of the test execution shall be housed in a specified directory as configured by the DevOps team. Supplementing this document is the **Test Plan Implementation – Technical Guide** which shall be a reference document for any test engineer starting on creating the test automation scripts.

III. Testing Strategy

The test strategy to be employed follows that of the Behavioral Development and Testing Strategy. Behavior Driven Testing (BDT) is the testing of the external behavior of a program, also known as block box testing or functional testing. The test scripts are created in such a way that it is in a language easily understood by the stakeholders and presents the testing in a story telling manner.

IV. Testing Tools

The test automation shall use the **cypress.io** test framework. This was selected due to it's ability to support fast, easy, and reliable testing scripts and built-in support for test document and deliverables such as videos and screenshots after every execution. It is likewise easily integrated with Node.Js and Javascript frameworks such as React.js. Any solid javascript developer can become a test engineer on his own by coding the automation test scripts in a natural programming environment with not much overhead to be able to begin creating scripts. The test scripts can easily be integrated with CI/CD tools as set up by the DevOps team. The monthly amortization computation uses the **financejs** javascript library to easily provide monthly payment amount by just providing the loan amount, period, and interest rate.

V. Test Execution

The automated testing coverage shall include the following:

1. **User Interface(UI) Validation** – Test scripts to validate the UI static data such as labels, internationalization I18n, font styles, forms, etc.
2. **Form Validation** – Test scripts to validate form element behaviors on certain events such as on load, on focus, on blur, on submit.
3. **Page Transitions and Events** – Test scripts to validate certain results on a call to action even like displaying the Amortization Schedule table upon form submit.

VI. Test Schedule

The automated testing could be triggered to do sanity testing by executing the sanity test scripts on each code commit to a provided GIT repository as part of the CI pipeline. Full testing execution which includes multi-browser testing may be configured by the DevOps to be executed every 11:59 PM on each day. Evidence of test execution will be stored in a configured directory for audit. Any regression result will be sent via email to the concerned developers and QA lead/s.