***TEST PLAN***

***DISCOUNT CALCULATOR***

***02/09/2022***

***• Introduction***

***This test plan describes the testing approach and overall framework that will drive the testing of the Calculator.net website.***

***1.1 Scope***

***In Scope***

***The test plan will only cover the discount calculator page for this website. All features on this page are part of the scope.***

***Out of Scope***

***All other pages are not in scope for this test plan.***

***1.2 Quality Objective***

***• Ensure the Application Under Test conforms to functional and non-functional requirements***

***• Ensure the AUT meets the quality specifications defined within the discount calculator specs***

***• All automation tests are ready and part of CI/CD to minimize the need to manually test the webpage***

***• Bugs/issues are identified and fixed before the web page is live.***

***1.3 Roles and Responsibilities***

***• Reshail: Create a test plan draft and automation framework for the website.***

***• Test Strategy***

***2.1. Test Objectives***

***The objective of the test is to verify that the functionality of discount calculator module for calculator.net website works according to the specifications.***

***The final product of the test is twofold:*** 

***• A production-ready software;***

***• A set of stable test scripts that can be reused for Functional and UAT test execution.***

***2.2. Test Assumptions Key Assumptions***

***• Production like data required and be available in the system prior to start of Functional Testing***

***• Exploratory Testing would be carried out once the build is ready for testing***

***• Performance testing is not considered.***

***• All the defects would be created on Jira with steps to reproduce and screenshots along with them.***

***• The Test Team will be provided with access to Test environment***

***2.3 Test Levels***

***Unit test: This will be conducted by the devs and sonarqube will be used to evaluate test coverage for every pull request. A minimum of 85% test coverage will be mandatory for any code change to be merged.***

***Smoke tests: All P0 test cases will be automated and executed as part of CI/CD for every pull request***

***Exploratory testing: The purpose of this test is to make sure critical defects are removed before the next levels of testing can start.***

***Functional testing: Once code changes are merged and deployed to a test environment assigned tester will perform functional testing only on the discount calculator for the website.***

***Regression test: All tests are to be executed as a nightly build on Jenkins and as part of every release.***   
  
***UAT: This will be performed by the tester allocated to the project to make sure all specifications are met and no gap in testing is found.***

***2.4. Test Principles***

***• Testing will be focused on meeting the business objectives, cost efficiency, and quality.***

***• There will be common, consistent procedures for all teams supporting testing activities.***

***• Testing processes will be well defined, yet flexible, with the ability to change as needed.***

***• Testing activities will build upon previous stages to avoid redundancy or duplication of effort.***

***• Testing environment and data will emulate a production environment as much as possible.***

***• Testing will be a repeatable, quantifiable, and measurable activity.***

***• Testing will be divided into distinct phases, each with clearly defined objectives and goals.***

***• There will be entrance and exit criteria.***

***2.5 Validation and defect management***

***Severity***

***Impact***

***Critical***

***• This bug is critical enough to crash the system, cause file corruption, or cause potential data loss***

***• It causes an abnormal return to the operating system (crash or a system failure message appears).***

***• It causes the application to hang and requires rebooting the system.***

***High***

***• It causes a lack of vital program functionality with workaround.***

***Medium***

***• This Bug will degrade the quality of the System. However there is an intelligent workaround for achieving the desired functionality - for example through another screen.***

***• This bug prevents other areas of the product from being tested. However other areas can be independently tested.***

***Low***

***• There is an insufficient or unclear error message, which has minimum impact on product use.***

***Cosmetic***

***• There is an insufficient or unclear error message that has no impact on product use.***

***2.5 Entry and Exit Criteria***

***Entry Criteria:***

***Once a build passes the unit tests, smoke test and is deployed to test environment testing can be started.***

***Exit criteria:***

***• 100% Test Scripts executed 95% pass rate of Test Scripts***

***• No open Critical and High severity defects 95% of medium severity defects have been closed All remaining defects are either canceled or documented as Change Requests for a future release***

***2.6 Test Completeness***

***• 100% test coverage***

***• All Manual & Automated Test cases executed***

***• All open bugs are fixed or will be fixed in next release***

***• Test Deliverables***

***• Test Plan***

***• Test Cases***

***• Requirement Traceability Matrix***

***• Bug Reports***

***• Test Strategy (including automation)***

***• Resource & Environment Needs***

***4.1 Testing Tools***

***• Test rail to be used as test case management tool***

***• Jira to be used for defect tracking and creation***

***• Automation tools: Selenium with docker***

***• Terms/Acronyms***

***TERM/ACRONYM***

***DEFINITION***

***UAT***

***User acceptance testing***

***AUT***

***Application under test***

**TEST CASES**

**Testcase 001: Verify the correct discount amount is calculated**

**Steps:**

**• Navigate to https://www.calculator.net/discount-calculator.html**

**• Select the `Percent off` radio button.**

**• Enter a valid value for `Price before discount` textbox**

**• Enter a valid value for `Discount` textbox**

**• Click Calculate**

**Expected result: The Result banner should be displayed followed by correct `price after discount` and `You saved` values.**

**Testcase 002: Verify Fixed amount off is calculated correctly**

**Steps:**

**• Navigate to https://www.calculator.net/discount-calculator.html**

**• Select the `Fixed amount off` radio button.**

**• Enter a valid value for `Price before discount` textbox**

**• Enter a valid value for `Discount` textbox**

**• Click Calculate**

**Expected result: The Result banner should be displayed followed by the correct `price after discount` and `Discount percentage` values.**

**Testcase 003: Verify clear button resets the values for Fixed amount off calculator**

**Steps:**

**• Navigate to https://www.calculator.net/discount-calculator.html**

**• Select the `Fixed amount off` radio button.**

**• Enter a valid value for `Price before discount` textbox**

**• Enter a valid value for `Discount` textbox**

**• Click Clear**

**Expected result: All textbox should be reset to default state**

**Testcase 004: Verify clear button resets the values for Percentage off calculator**

**Steps:**

**• Navigate to https://www.calculator.net/discount-calculator.html**

**• Select the `Fixed amount off` radio button.**

**• Enter a valid value for `Price before discount` textbox**

**• Enter a valid value for `Discount` textbox**

**• Click Clear**

**Expected result: All textbox should be reset to default state**

**Testcase 005: The user should be able to search for discount calculator**

**Steps:**

**• Navigate to https://www.calculator.net/discount-calculator.html**

**• Click on the search bar on the right side of the page**

**• Search for `discount calculator`**

**• Click `Discount calculator` from recommendation**

**Expected result: The user should be navigated to the discount calculator page**

**Testcase 006: The description for discount explanation should be visible**

**Steps:**

**• Navigate to https://www.calculator.net/discount-calculator.html**

**• Scroll to discount description**

**Expected result: The discount description and calculation example should be visible to the user**

**Testcase 007: The user is able to navigate to different financial calculators using available shortcuts**

**Steps:**

**• Navigate to https://www.calculator.net/discount-calculator.html**

**• Click any of the options (example Mortage)**

**• Use should be redirected to the selected calculator**

**Expected result: The user should be redirected to the correct URL and the page should not crash.**

**Testcase 008: The calculator should not allow negative values to be used**

**Steps:**

**• Navigate to https://www.calculator.net/discount-calculator.html**

**• Select the `Percentage off` radio button.**

**• Enter a negative value for `Price before discount` textbox**

**• Enter a negative value for `Discount` textbox**

**• Click calculate**

**Expected result: User should see the following error messages**

***Please provide a positive price before discount value.***  
***Please provide a positive discount value.***

**Testcase *009: User should not be able to enter discount value > 100***

***Steps:***

**• Navigate to https://www.calculator.net/discount-calculator.html**

**• Select the `Percentage off` radio button.**

**• Enter a valid value for the `Price before discount` textbox**

**• Enter a value greater than 100 for `Discount` textbox**

**• Click calculate**

***Expected result: User should see the following error message***

***Please provide a discount value of less than 100***

**Testcase *010: `Price after discount` and `you saved` fields should not be editable***

***Steps:***

**• Navigate to https://www.calculator.net/discount-calculator.html**

**• Select the `Percentage off` radio button.**

***• Edit `Price after discount` textbox***

***• Edit `You saved` textbox***

***Expected result: Both textboxes should not be editable***

**Testcase *011: `Price after discount` field should not be editable for fixed amount off calculator***

***Steps:***

**• Navigate to https://www.calculator.net/discount-calculator.html**

**• Select the `Fixed amount off` radio button.**

***• Edit `Price after discount` textbox***

***Expected result: The price after discount textbox should not be editable***

**Testcase *012: The user is able to use the breadcrumbs to navigate back from the discount calculator***

***Steps:***

**• Navigate to https://www.calculator.net/discount-calculator.html**

**• Select the `Financial` from the breadcrumb link(s)**

***• Select `Home` from the breadcrumb link(s)***

***Expected result: User should navigate to the correct page successfully***