**Test Plan**

**Test Objective:** Build an automated test suite that tests the functionality of the ‘calculator’ app on Google Home Page. Note the calculator app appears dynamically as a search result and is published on the webpage. The automated tests must interact with this dynamic web app.

**Test Approach:** Since only the numeric buttons **(0 1 2 3 45 6 7 8 9),** non-numeric buttons **(**. =+-×÷) and clear buttons (AC and CE) are required to be tested, the test script will have 4 functions. Python will be used to code the test scripts. Selenium will be used as the WebDriver. Page Object Model test design pattern will be used. As a result there will be a ‘page’ python file that contains the functions and a ‘test’ python file that contains the test cases.

**Test Setup:** A laptop running Windows 10 is sufficient to code the tests and execute them. Google Chrome will be used as the web browser. The calculator app appears on the web browser Google home page when ‘calculator’ is used as the keyword and searched. The calculator app is dynamic and the tests must interact on this app.

**Test Steps:**

1. Identify the build version of Chrome web browser by looking at the ‘About Chrome’ page under ‘Settings’.
2. Download the latest Chrome driver for Selenium, extract the zip file and store chromedriver.exe in C:\Drivers\chromedriver-win64\chromedriver.exe; Set this as the PATH variable in Environment settings of Windows System control panel app.
3. Install Python; my version is 64 bit Python 3.11.0; set the PATH variable.
4. Using Visual Studio Code Editor IDE create Python page.py file and test.py file.
5. Install Selenium using **Pip** command. Install the page module.
6. Code the Python import statements in the page.py and test.py files.
7. Create a class in the test.py file with ‘def’ functions for **def setUp(), def test\_button(),**

**def tearDown()**

1. Make test.py a python executable script by adding the statement in the end **if \_\_name\_\_ == “\_\_main\_\_”:**
2. In the page.py file create functions using ‘def’ keyword for **def AC\_button(), def numeric\_button(), def math\_button() and def CE\_button().**
3. Check-in code into GitHub repository.

**Test Execution:**

1. Run test.py script from Visual Studio Code Editor IDE.
2. Let the test.py script click ‘AC’ button and verify the display is 0
3. Let the test.py script click ‘8’ numeric button and verify display is 8
4. Let the test.py script click ‘+’ button, ‘9’ button and ‘=’ button and verify display is the sum 17
5. Let the test.py script click ‘9’ button, clear it using ‘CE’ button and verify display is 0

**Test Results in ZIP file:**

1. File ‘Screenshot\_google.png’ shows display of 8 when test script clicks button ‘8’
2. File ‘TestResult\_AutoDeskPOM.docx’ shows verification results when test.py script is executed with results displayed in Visual Studio Code Editor IDE console.

**Code Deficiencies:**

1. Please note following buttons were not coded and tested: **0 1 2 3 4 5 6 7 and (**. -×÷)
2. Reason is that instead of repeating coding of functions for each of these buttons a test approach known as **test parameterization** is required. I did not have time to do test parameterization of repeat test data. It involves setting up a test framework for this.

**Contents of ZIP file:**

1. Test Plan.docx

2. Screenshot\_google.png

3. TestResult\_AutoDeskPOM.docx

4. AutoDeskPOM folder containing page.py and test.py code files

**My Visual Studio Code Editor Setup:**

