Contents

[Objective: 1](#_Toc64049891)

[Test Approach 1](#_Toc64049892)

[Technology Stack: 1](#_Toc64049893)

[Test Coverage: 2](#_Toc64049894)

[Automation Solution 2](#_Toc64049895)

[Automation tool: 2](#_Toc64049896)

[Framework: 2](#_Toc64049897)

[Reporting: 2](#_Toc64049898)

[Solution Implementation 2](#_Toc64049899)

[Execution Guide 3](#_Toc64049900)

[Software Required: 3](#_Toc64049901)

[Tool Download and Setup: 3](#_Toc64049902)

[Test Data setup: 3](#_Toc64049903)

[Test Execution: 3](#_Toc64049904)

[Execution Results: 4](#_Toc64049905)

[Results Analysis: 4](#_Toc64049906)

# Objective:

This program executes the automation test scripts of search functionality of Google search engine/home page based on the parameterized input provided in configuration test file by user and generates a html reporting with validations.

# Test Approach

## Technology Stack:

* Java
* Selenium
* TestNG
* Extent Reporting
* Maven
* Git

## Test Coverage:

The test suite covers the basic functionality of searching and validating the search result links of any provided keyword.

# Automation Solution

## Automation tool:

Selenium with Java is used as the testing tool as the test scenarios are purely web UI automation and no non-web application interactions are involved for the selected tests.

## Framework:

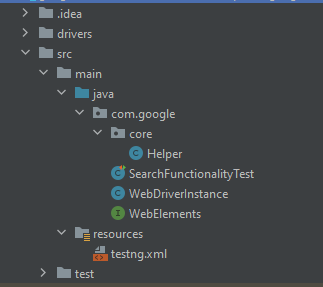
TestNG with Maven is implemented as core framework considering the advantages provided by the annotations and also TestNG is preferable to select in scenarios where the test data is minimal and rate of changes/renewal is also minimal. Also finally, TestNG accommodates the automation test if in case the size of project increases with minimal efforts in framework changes.

## Reporting:

Extent reporting is used as reporting framework to provided user friendly web based/html reporting.

# Solution Implementation

The solution includes three classes and one Interface in order to achieve the required automation solutions shown in below screenshot of IDE.



# Execution Guide

## Software Required:

* Chrome browser 87
* Maven
* JDK 1.8

## Tool Download and Setup:

Download the tool from below git repository location

Repo: <https://github.com/PhaniKonda/SeleniumGoogleSearch>

Branch: main

Clone the above repo in to any location in local machine which is from now considered are root directory or project working directory.

## Test Data setup:

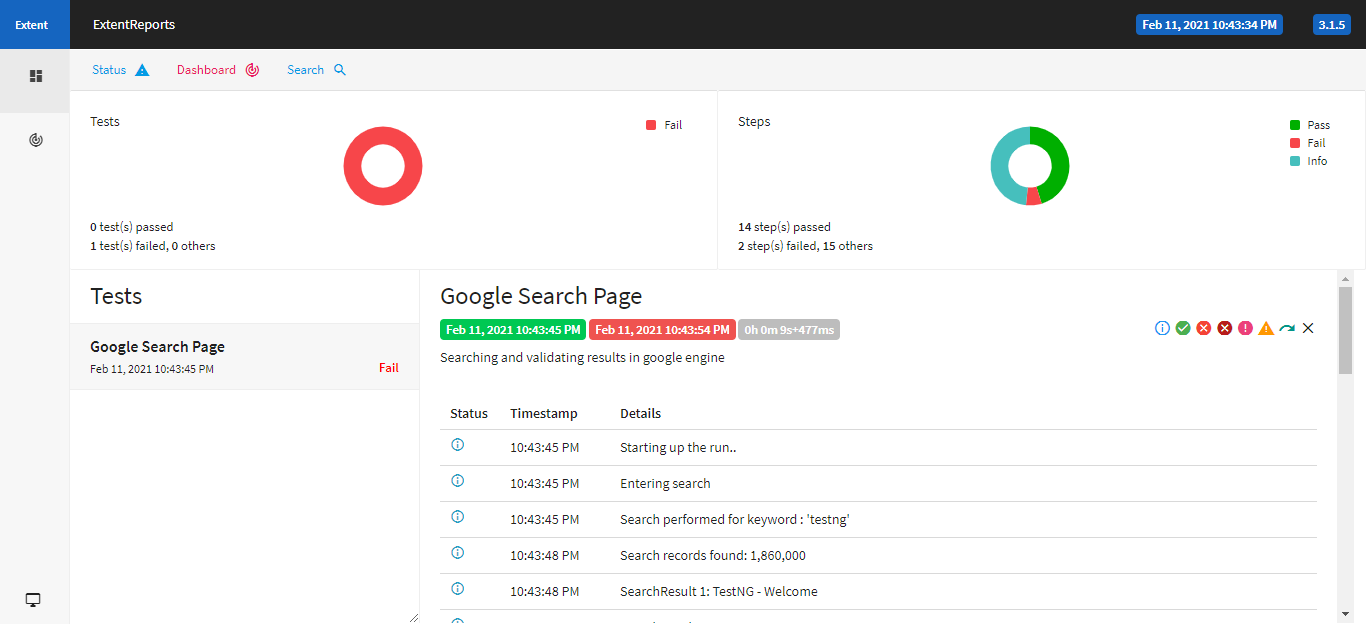
Test data can be altered or changed as per the end user requirement in testng.xml file present at resources folder “./src/main/resources/testng.xml”. Currently the test data will search for keyword “Selenium” and validates the search results.

## Test Execution:

Double click the “runner.bat”

## Execution Results:

Once the execution completes the tool will generate an html file named “results.html” at root directory or project working directory. Double click the file to open the results which looks like something shown below.



## Results Analysis:

A very nice detailed reporting is already provided by extent reporting as shown above which communicates best possible way of steps performed while the test execution running any failures will be marked with red as shown below.

