# Test Plan

## Search Feature

Contents

[1. INTRODUCTION 1](#_Toc87887419)

[2. TESTING STRATEGY 1](#_Toc87887420)

[a. Test Scenarios 1](#_Toc87887421)

[b. Assumptions: 2](#_Toc87887422)

[3. Automation solution 2](#_Toc87887423)

[a. Tools Required 2](#_Toc87887424)

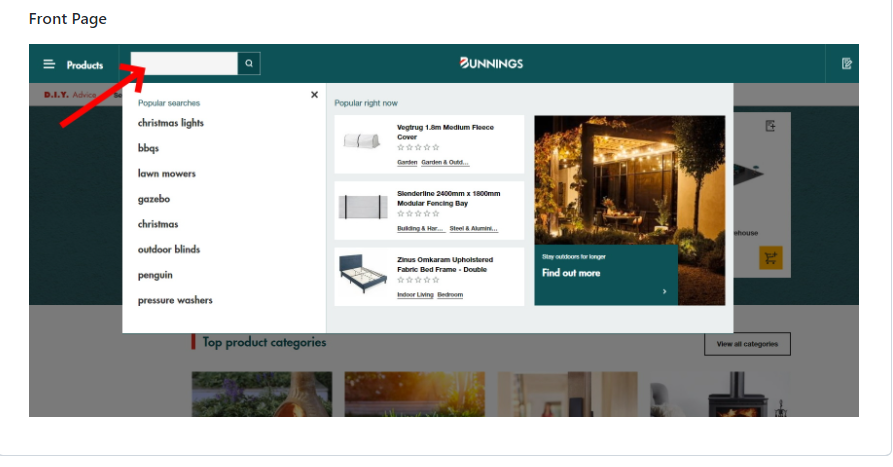
[b. Framework Set up 2](#_Toc87887425)

[c. Framework structure 2](#_Toc87887426)

[d. Report 3](#_Toc87887427)

# INTRODUCTION

This document covers test plan and test scenarios around search feature in Bunnings website and automation solution to set up a regression suite to test the same.



# TESTING STRATEGY

In order to set up a regression suit we can follow below actions:

1. identify the scenarios around search bar.
2. Test data set up
3. Automation Tools installed
4. Project set up
5. Link with Jenkins and create a pipeline, we can run the pipeline to check on regression test cases as well to identify any new feature in case of any failures.

## Test Scenarios

|  |  |
| --- | --- |
| # | Test Scenario |
| 1 | Verify that search field is displaying on home page |
| 2 | Verify that user is able to edit the search bar |
| 3 | Verify the search should not accept any special chars |
| 4 | Verify that user is able to search the product via search bar |
| 5 | Verify that flyout should open on search bar click |
| 6 | Verify that User should be able to select random item from flyout |
| 7 | Verify that popular searches should display in fly out |
| 8 | Verify the recent searches should display in fly out in case of same session |
| 9 | E2E\_Verify that guest user is able to search, add and order the item |
| 10 | E2E\_Verify that registered user is able to search, add and order the item |

## Assumptions:

Data used in scripts should be stable for example if I am hitting sofa in search bar , result count should be stable, so always better to use specific product id rather than giving a generic product name.

# Automation solution

We can use Selenium webdriver based framework to develop the regression suit for above scenarios.

## Tools Required

Build Tool – Gradle 7.2

Automation tool – Selenium Webdriver 3.141, Cucumber (It can be downloaded via build file)

Programming language – JDK 8 or above

IDE – Intellij IDEA

Version control – GIT

## Framework Set up

System should have all the above mentioned tools in order to set up a framework. Once a gradle project is initiated, it will auto generate the build.gradle file. This is written in groovy and a first point to trigger the execution.

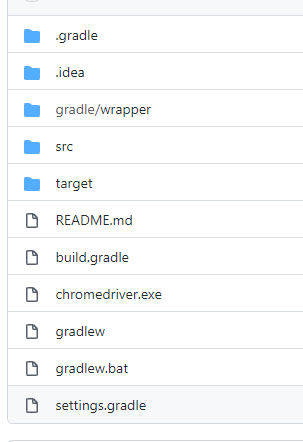
In build.gradle file we add multiple tasks to start the execution also we add different dependencies like selenium, junit and cucumber to install in project folder.

Once the dependencies are all downloaded and project is build, it will auto create the src folder. Two main folder reside inside src folder main and test. These are the place where we can add resources, java classes, Object library etc.

We can create a repository in git for source code management

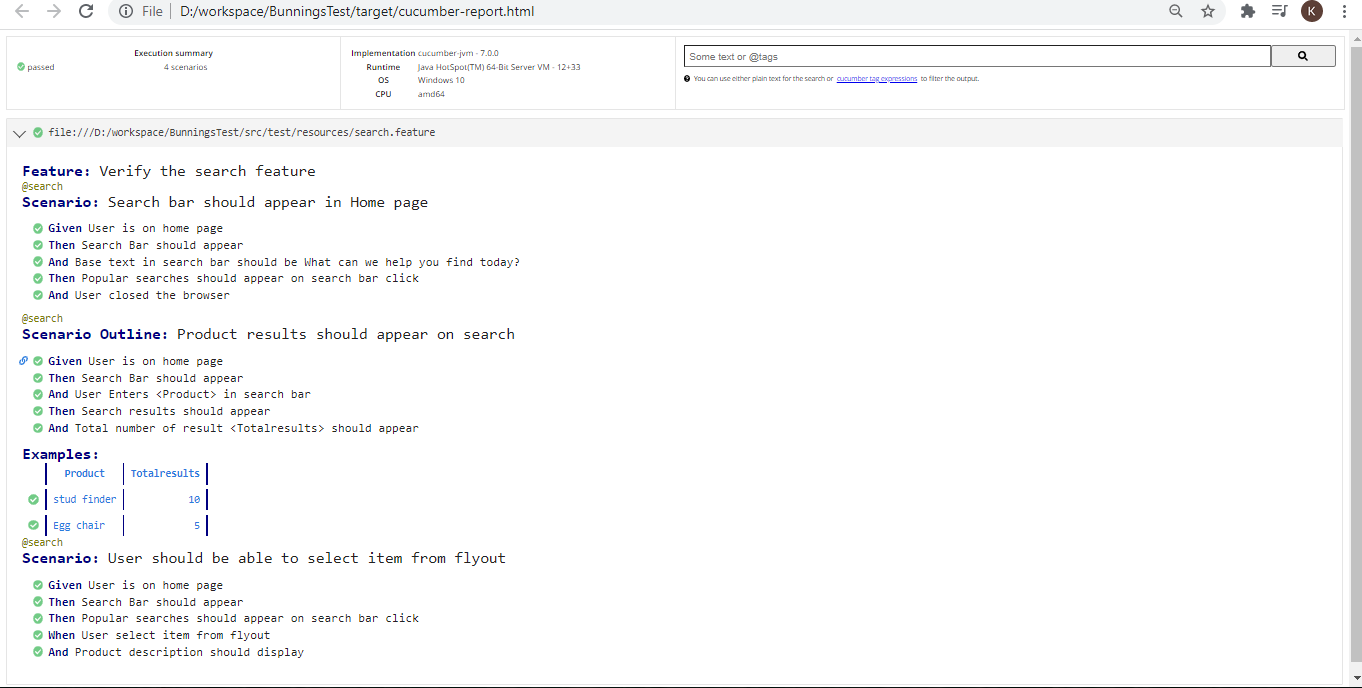
## Framework structure

This Framework structure look like this -



## Report

After the execution cucumber report will generate in target folder. There are many different plugins can be used to generate different reports.



Please refer repo <https://github.com/kritika333/BunningsTest> for more details.