# 

# Test Plan Document

Project - Challenging DOM

**Introduction**

The Test Plan has been created to define and communicate the Test approach, to Stakeholders and team members. It includes the Test objectives, Testing scope for application under test, Assumption, Automation Test approach, risks and Test deliverables. This document will clearly identify what the test deliverables will be and what is deemed in and out of scope.

## Objectives

Given webpage /Application URL ‘ <https://the-internet.herokuapp.com/challenging_dom> ’ contains the different webpage elements and it is created to perform all the possible tests successfully identify those all elements C (e.g., ones that well named, unique, and unlikely to change.)

The web application is given as an exercise / Test to the candidate to write possible Automated Tests which cover all the functionality of the that webpage and as per given assumption. A minimum of ten automated tests is expected.

## Test Team /Tester

|  |  |
| --- | --- |
| **Resource Name** | **Role** |
| Neelam Rathi | T Test Analyst |

## Testing Scope

Application Under Test page / URL –

‘ <https://the-internet.herokuapp.com/challenging_dom> ’

The automated tests must be created using open source tools, languages and frameworks.

The candidate should then produce a set of automated tests to test the web application.

A minimum of ten automated tests is expected. The automated tests must be freely available on open source tools, languages and frameworks

* The Automation Test Set should cover-

Locate the below webpage element unequally and perform and validate the possible action.

* + - Button
    - Image link Text
    - Link Text
* Locate the webpage Table and read the table Row & Column values, Headers, identify the Index.
* Print the Table & all displayed values.
* Locate the table link text and perform the desired function.
* Locate & read the canvas element Test values to validate the page refresh and action on colored button.

## Assumptions-

1. The web application under test should be stable. – ‘ <https://the-internet.herokuapp.com/challenging_dom> ’
2. All the given functionality will be available / delivered on the planned deployment date in Test environment.
3. These colored buttons should be visible & enable.
4. On clicking those buttons webpage gets refreshed and should display new number text in the image box (canvas element).
5. The link Text ‘[Elemental Selenium](http://elementalselenium.com/)’ should open a new web page with header ‘Elemental Selenium’ in new browser tab when clicked.
6. Image link Text ‘Fork me on GitHub’ should navigate to GitHub repository in the same tab.
7. The given Web table has 10 rows and 7 columns with header.
8. The link ‘edit delete’ displayed under Action Column for each row.
9. While clicking ‘edit’ and ‘cancel’ of any row should navigate to the following respective URLs –‘ <https://the-internet.herokuapp.com/challenging_dom#edit> ’ , ’ <https://the-internet.herokuapp.com/challenging_dom#delete> ’

## Automation Test Approach – Framework & Tools .

To Automate all the functionality as per Test scope & Assumption BDD test Automation framework will be created used.

### Test Framework – BDD

BDD framework i.e. Behaviour Driven Development is a software development approach that allows the tester/business analyst to create test cases in simple text language (English).

The simple language used in the scenarios helps even non-technical team members to understand what is going on in the software project. This helps and improves communication among technical and non-technical teams, managers, and stakeholders.

* Test scenarios are written separately in a different file, named as Feature file.
* Tests are written by focusing user stories and system behavior in a layman language.
* Code is subject to be written differently in step definitions file i.e. Java, Python.

All the following Tools / Technologies will be used in Automation framework

|  |  |
| --- | --- |
| **Tools/Technology** | **Purpose** |
| IntelliJ | integrated development environment (IDE) -It will provide the platform and environment to write & execute our Automation Test code |
| Cucumber | Test framework that supports BDD, BDD specifications are written in plain, simple English which is defined by the Gherkin language |
| Gherkin | Cucumber specific language which helps you to describe business behaviour, write Scenarios in simple English. |
| Selenium Web driver | Web Automation Tool, It is a collection of open source APIs and will be used in our Testing Framework to automate he User interaction with web application |
| Java | Programming language to write the automation test code |
| Maven | Project/ Build management tool, It is based on the concept of a POM (Project Object Model) contains project information and configuration information. It will be used to build the project like build directory, source directory, dependencies, test source directory, plugin, test reports etc |
| Junit | Testing framework, will be used to run the tests and perform test validation, assertion etc. |
| Java Script | Scripting language |

## Test Coverage-

* Automation tests will be run to verify the functionality of web application on Chrome Browser.
* Test will be performed on the following chrome browser version – ‘79.03945.117’
* The Test will be run on the Windows 10 operating system.
* The current Test scope is to perform the function testing (no other testing is in scope like - performance).

## Test Schedule-

* Automated Test Framework & Test Case Preparation – 24hrs prior to schedule Interview
* Automated Test Cases Execution – During the schedule interview

## Test Scenarios & Test Cases



## Test Deliverables

1. Test Plan Document
2. Test scenarios & Automated Test Cases
3. Automated Test Framework including test script.
4. Test Execution Reports.