

# Test Strategy & Plan

This document specifies both test strategy and planning used to test the Challenging DOM application.

## 1. Introduction and Scope:

- The purpose of the challenging DOM app is to show how can we use different locators such as id, name, xpath, css-selector etc. to find the same element on a webpage.
- Both static and dynamic locators are present in the challenging DOM application.
- The scope of this project is to test the basic functionality of application by uniquely identify both static and dynamic elements using advance locator techniques.

## 2. Test Approach:

This section specifies types of testing that will be conducted and environment specifications.

### 2.1 Types of testing:

We will be conducting exploratory testing and automated functional testing.

- Exploratory testing helps to understand the behaviour of the application.
- Test automation is used to automate the functionality.

### 2.2 Environment Specifications:

- Windows 10 laptop.
- Chrome browser

## 3. Testing tools:

List out various tool used for testing challenging-DOM.

- Selenium WebDriver: Web automation tool.
- TestNG: Unit Testing
- Eclipse: IDE
- Maven: Build tool.
- Core Java: To write test scripts

## 4. Test items:

- There are three buttons on webpage that execute a functionality on clicking. The locator (eg. id) of these buttons are dynamic in nature. It changes when we click either of the three buttons.
- The name on each of the buttons also change every time we click it.

## **5. Features to be tested:**

1. Verify the application page title.
2. Verify if the name of the application - Challenging DOM is displayed on page.
3. Verify if the description of Challenging DOM application is displayed on page.
4. Display all values from table in tabular form.
5. Verify the presence of 'Fork me on GitHub' logo.
6. Identify dynamic locators of three buttons using dynamic xpath.
7. Verify if all three buttons are enabled.
8. Verify clicking of three buttons using dynamic xpath.

## **6. Test Execution:**

To run automation scripts:

- Set up the environment using tools described in section 3.
- In project, Right click on file (testng.xml) > Run-as > TestNG Suite.

## **7. Test Deliverables**

- Test strategy and plan document.
- Test scenarios
- Test scripts (Location in project folder:  
ExpleoAutomationTest\src\com\test\project\ChallengingDOM.java)
- Test Execution Report (Location in project folder: ExpleoAutomationTest\test-output\index.html)