**Project Overview**

This document explains about the automation project that uses playwright with typescript as a tool to execute some sample tests.

**Overview of the project:**

* **Playwright:** Playwright is used in this project to execute the tests on browser. Playwright is an open-source automation library for browser testing. Its known for speed, modern architecture, and built-in features like auto-waiting and fast execution.
* **TypeScript:** typescript is used as a language to write the code. Playwright supports typescript in every aspect of the automation.
* **POM:** project framework used page object model for better code structure and maintainability in the future. Every UI page has a separate class for methods and locators for ease of implementation of test cases. It will help to manage the project in modules which increase the readability of the project.
* **Allure:** Allure reporting tool is used in the project for the best view of results on the UI. Allure is a third party open-source tool integrated with the playwright.
* **Test Data:** No hardcode values are used in the project; all the test data is passed through .json files and used.
* **Multi-browser:** User can run this project on any browser. In the playwright.config.ts file have the configuration for all the browses.
* **Assertions:** Playwright uses the expect library from Jest for test assertions, providing matchers like toBeChecked, toBeVisible, toContainText, toHaveText, etc., to validate conditions during automated tests.
* **Helper:** Helper folder will have the classes for common methods that can be used throughout the project.
* **Documentation:** project structure and initial setup of the project, how to run the project, generate the allure report and view is explained the documentation in the documents folder of the project.
* **Configurations:** Mos the project configurations can be manage from the playwright.config.ts file in the project where user can manage – browsers configuration, test directory, retires, reporting, on failure screenshots etc.

**Note:** This project uses ‘https://www.douglas.de/de’ for testing. Where below test is scripted:

**Test Case:** Verify aktionen filters on Parfum page

**Test data:** Sale, NEU, Limitiert

**Steps:**

* Navigate to the url
* Accept the cookies consent
* Click on Parfum Tab and wait for tab selected
* Click on actioned dropdown and select filter