# Introduction:

This project is an automation framework for Flipkart, an e-commerce platform. It contains test cases for adding products to the cart, searching and sorting for products, and searching flights.

The framework utilizes Selenium WebDriver for web automation, TestNG for test execution and reporting, and maven as build execution tool.

**Website Used**: [Flipkart](https://www.flipkart.com/)

# Tools Used:

* Automation testing tool - Selenium WebDriver 4.9.0
* Programming language – Java 16.
* Logging - Log4J2 2.2.0
* Design Pattern - Page Object Model (With Page Factories)
* Testing framework – TestNG 7.8.0
* Build and Dependency Management Tool – Maven 3.9.3.

# Project Structure:

The project follows a Maven project structure with page object model framework. The main packages and files are as follows:

1. **com.flipkart.automation.pages:**
   * **BasePage.java**: Base class for page objects, providing common methods
   * **HomePage.java**: Page object class for the home page, containing methods related to home page actions.
   * **ProductListPage.java**: Page object class for the product details page, containing methods for product lists and sorting.
   * **ProductDetailsPage.java**: Page object class for the product list page, containing methods for product details and adding to cart.
   * **CartPage.java**: Page object class for the cart page, containing methods for cart-related actions.
   * **SearchFlightPage.java**: Page object class for the flight booking page, containing methods for searching flights.
2. **com.flipkart.automation.tests:**
   * **BaseTest.java:** Base class for test classes, initializing WebDriver and setting up test configuration.
   * **AddProductToCartTest.java:** Test cases for adding products to the cart, verifying titles, and product count in the cart.
   * **SearchFlightTest.java:** Test cases for searching flights, entering cities, and verifying search results.
   * **SearchProductByCategoryTest.java:** Test cases for searching products by category, and verifying search results.
   * **SearchProductTest.java:** Test cases for searching flights, entering cities, and verifying search results.
   * **SortProductTest**.**java**: Test cases for sorting based on price.
3. **com.flipkart.automation.utils:**
   * **ExcelFileReader.java:** Utility class for reading test data from excel file**.**
   * **ExtentReporterConfig.java: U**tility class for configuring extent reports for test reporting.
   * **JsonFileReader.java:** Utility class for reading test data values from json files.
   * **PropertyFileReader.java:** Utility class for reading property values from the GlobalData.properties file.
   * **ReportUtility.java:** Utility class for implementing the testng listeners, taking screenshot om failure, adding logs to the extent report.
   * **StringFunctions.java:** Utility class for string manipulation and directory operations**.**
4. **testngSmoke.xml**: TestNG configuration file for smoke test suite execution.
5. **testngRegression.xml**: TestNG configuration file for regression test suite execution.
6. **testngGrouping.xml**: TestNG configuration file for groups test suite execution.
7. **Current test results**: To store the report of latest executed test cases.
8. **Archive test results**: To store the reports of previously executed test cases.

# Usage Instructions to run through IDE

1. Clone the project repository to your local machine as Maven project.
2. Execute Maven->Update command to install the required libraries.
3. Download the latest TestNg version through Eclipse marketplace and restart.
4. Install the java version greater than 11 to support latest TestNg version.
5. Update the GlobalData.properties file with relevant test data and configurations.
6. Update the testng.xml files if require to have the test cases.
7. Run the TestNg files by right clicking on xml file and selecting RunAs-TestNgSuite.
8. Refresh the project and check the test reports in Current test results folder with timestamp.

# Usage Instructions to run through Terminal

1. Clone the project repository to your local machine as Maven project.
2. Install the maven version 'apache-maven-3.9.3' in your machine.
3. Install the java version greater than 11 to support latest TestNg version.
4. Add maven and java path to Environment variables or bash\_profile as per your system.
5. Update the GlobalData.properties file with relevant test data and configurations.
6. Update the testng.xml files if require to have the test cases.
7. Navigate to pom.xml file path through terminal.
8. Run command **'mvn test'** to run the default smoke test cases present in testng.xml file.
9. Run command **'mvn test -P CartTestCases'** to run test cases based on grouping.
10. Run command **'mvn test -P Smoke'** to run smoke test cases.
11. Run command **'mvn test -P Regression** to run all test cases.
12. Modify the profile names if required.
13. Check the test reports in Current test results folder with timestamp.

**Note**: Ensure that the test environment has an active internet connection for accessing the Flipkart website.

# Additional Notes:

1. The framework uses the Page Object Model (POM) design pattern for better maintainability and reusability.
2. Extent Reports are generated for test reporting, providing detailed test execution results.
3. Custom utility classes are implemented for string manipulation, property, Json, excel file reading and directory operations.

For any further questions or issues, please contact the project maintainers.

Maintainers:

[Sukant]