* **Algorithm**

Phase-End Project

Automate an E-Commerce Web Application

1. Set up the necessary dependencies:
   * Create a new Maven project and add the required dependencies (Selenium, TestNG, and ReportNG) to the pom.xml file.
2. Create TestNG Test Class:
   * Create a new Java class (e.g., FlipkartLazyLoadingTest) and annotate it with @Test annotations for different test scenarios.
3. Implement Test Scenarios:
   * Inside the FlipkartLazyLoadingTest class, implement the test scenarios based on the provided code, including:
   * Test the page load time.
   * Test the product search and lazy loading of images.
   * Test the content refresh frequency while scrolling.
   * Test that different browsers and screen resolutions render the application the same way.
4. Create testng.xml file:
   * Create a testng.xml file to define the test suite and test classes to be executed.
5. Set up WebDriver and Browser Configuration:
   * In the setUp() method of the test class, initialize the WebDriver (e.g., ChromeDriver) and set up browser configuration (e.g., maximize window, implicit wait).
6. Execute the Test Suite:
   * Right-click on the testng.xml file and run it as a TestNG Suite. This will execute all the test cases defined in the test class.
7. Generate Reports:
   * After test execution, ReportNG will generate HTML reports with test results, including passed and failed test cases.

# Sprint Planning

* **Sprint Duration:** 2 weeks (10 working days)
* **Sprint Goal:** Automate key functionalities of the e-commerce web application to ensure its end-to-end functionality, performance, and compatibility.

# User Stories / Tasks:

* **User Story:** As a Test Engineer, I want to verify the page load time of the e-commerce website.
  + Create a TestNG test case to measure the page load time.
  + Set up WebDriver and perform the test on different browsers and screen resolutions.
* **User Story:** As a Test Engineer, I want to test the product search and lazy loading of images feature.
  + Create a TestNG test case to search for a product and verify the loaded images.
  + Implement logic to check the lazy loading behavior of images.
* **User Story:** As a Test Engineer, I want to verify the content refresh frequency during scrolling.
  + Create a TestNG test case to simulate scrolling and check content refresh frequency.
* **User Story:** As a Test Engineer, I want to ensure that different browsers render the application consistently.
  + Set up WebDriver configurations for multiple browsers (e.g., Chrome and Firefox).
  + Run the existing test cases on different browsers and ensure consistent behavior.
* **User Story:** As a Test Engineer, I want to generate HTML reports for test results.
  + Integrate ReportNG into the project to generate HTML reports after test execution.

# Sprint Backlog / Task Breakdown

* + **Day 1-2:** Set up the Maven project, add dependencies to the pom.xml, and create the FlipkartLazyLoadingTest class.
  + **Day 3-4:** Implement the page load time test case and configure WebDriver for different browsers.
  + **Day 5-6:** Implement the product search and lazy loading of images test case.
  + **Day 7-8:** Implement the content refresh frequency test case and run tests on multiple browsers.
  + **Day 9:** Integrate ReportNG for HTML reporting and finalize any pending tasks.
  + **Day 10:** Perform a thorough review, execute test cases, and ensure all requirements are met.