TESTNG & JUNIT

1. **What is TestNG?**
   - You have 500 test cases→ We create a Java Package and 500 Class for each test cases
     Client asked you run only 40 of them for smoke test → We handle it in Jasmine with it blocks and reporting mechanism.
   - TestNG is a testing framework
   - Centralized controller: manage run different test cases then create reports, logs
   - Batch execution: 100 test cases and run them one by one
   - Optional execution: we can skip some test cases

2. **What is assertions in TestNG?**
   - We run the test and title test case failed. It will not affect the other test cases, so we don't want our script to stop.
     - Critical → stop/failure Assert
       - It takes one boolean argument and String message. It Asserts that a condition is true. If it isn't, an AssertionError, with the given message, is thrown.
     - Non critical → failure/continue SoftAssert
       - Soft Assert does not throw an exception when an assert fails and would continue with the next step after the assert statement.

3. **Difference between JUnit and TestNG**
   - Annotations; **JUnit:** @Test, @BeforeClass, @AfterClass, @Before, @After, @Ignore
     TestNG: @Test, @BeforeTest, @BeforeClass, @BeforeSuite, @BeforeMethod, @AfterTest, @AfterClass, @AfterSuite, @AfterMethod
   - Both are testing framework to help us running automation scripts.
   - TestNG provide html report
   - TestNG has @Dataprovider annotation same as Cucumber Scenario Outline for Data Driven Testing.
   - In TestNG, we can do parallel testing, but JUnit doesn't support to parallel test, so we use sauceLab for it.
   - TestNG support group test but JUnit doesn't support
   - TestNG and JUnit both of them have parameterize testing but TestNG parameterized test configuration is very easy to configure. There are two ways to achieve parameterization in TestNG;
     - @Parameters and TestNG xml file
     - @DataPrivider

FEATURE	JUNIT	TESTNG
Purpose	General unit testing	Focus on Integration testing for Enterprise projects
IDE support	yes	Yes
Maven support	yes	Yes
setup/teardown for test	@Before / @After	@BeforeMethod / @AfterMethod
setup/teardown for class	@Before / @After	@BeforeClass / @AfterClass
setup/teardown for suite	no	@BeforeSuite / @AfterSuite
setup/teardown for test groups	no	@BeforeGroups/ @AfterGroups
setup/teardown for test	in annotations	In annotations and/or XML file
Parameterised tests	Yes, but in a limited way	Yes
Test groups	Yes with Categories (new feature)	Yes
Test for Exceptions	Yes	Yes
Timeouts in tests	Yes	Yes
Test order	Non-Deterministic or alphabetical	Can be defined in detail with dependencies
Dynamic test input	No	Yes with DataProviders
Can run tests of the other library	No	Yes, TestNG can run JUnit tests
Assumptions before running a test	Yes	No
Dependency injection for tests	No	Yes, with Google Guice
Ignore/disable test	Yes	Yes
Parallel testing	No	Yes
Test listeners	No	Yes
Test reporters	No	Yes

4. **Cross Browser and Parallel Test**

- In my current project, we use sauceLab for cross browser testing. But my previous project I used testng.xml file.
- Basically, inside the suite there are 3 keys (name, thread count, parallel) and I created 2 different tests, one of them is for
- Chrome and the other one is for Firefox.
- There is also parameter annotation and include name and value; name is browser and value is Chrome.

```
<?xml version="1.0" encoding="UTF 8"?>
<!DOCTYPE suite SYSTEM ...>
<suite ...>
   <test name="ChromeTest" ... >
            <parameter name="browser" value="chrome"/>
                     <classes>
                              <class name="testsuite..."/>
                     </classes>
   </test> <!   First Test   >
   <test name="FireFox" ... >
            <parameter name="browser" value="FireFox"/>
                     <classes>
                              <class name="testsuite…."/>
                     </classes>
   </test> <!   Second Test   >
</suite> <!   Suite   >
```