Auden Assignment: Selenium Java

Goals/ Focus on

1. Creating project from Scratch –
   1. No code reuse from previous
   2. Project might be simpler and smaller
2. Meet as many specified test conditions as possible
   1. 3 test assertions specified
   2. Have to move the slider only
3. Finish by target of 3 Hours (total work time) . Actual time taken 5 Hrs

Tool Choice:

Framework: Selenium Java framework from scratch using TestNG, Extent reports,

IDE: Eclipse

Tool Choices reasoning

1. Comparatively easier to create projects from scratch.
2. Free. No Commercial license required
3. As good as any other tool for the task at hand
4. Can be easily extended to cover more advanced/extended features like data driven, BDD, Continuous Integration, etc.

Step by Step Approach:

Assuming a brand-new environment with working internet connection as the only pre-requisite

1. Download and install Java.
2. Download and install Eclipse IDE.
3. Install Maven integration for Eclipse (If this was not chosen during installation).
   1. Install TestNG plugin from market place this is essential if we need to run the tests as TestNG tests. If the TestNG JARs are installed, We will be able to use all the annotations. But to run as TestNG, the plugin needs to be installed from Eclipse Market Place. Good to refresh on this knowledge. Learnt this the hard way this time.
4. Create Maven Project
5. Visualise the simplest form of the framework required for the task.
6. Create organise folders and packages according to the framework visualisation and the rules set for the framework.
7. Modify the POM.xml to import the dependencies like selenium repositories, extent report, log4j, etc based on need.
8. Create separate packages to segregate PageObjects and TestCases. This is to give structure and discipline to the framework
9. Optional Step: Create a manual test case. This helps in identifying the steps and objects required for the task. This was skipped as this was a non-mandatory step and the test itself was functionally simple
10. Create page objects for the objects that we will be interacting with based on the test case and conditions. This is created as a separate class file under the PageObjects package.
11. Create test cases as classes in the testcases packages. This will the actions and assertions. Basically, this should be equal to the actions and expected results (assertions) of the normal manual test case.

Issues Faced:

1. Unable to find any page objects. org.openqa.selenium.NoSuchElementException: no such element: Unable to locate element:
   1. tried Firefox driver instead of chrome -that was not the problem.
   2. **RESOLUTION**: My conditional wait statement was faulty **switched to implicit wait** before throwing the NoSuchElementException
2. Slider Object Manipulation:
   1. No Handle found – Still works
3. StaleElementException : Calculated loan value
   1. Elegant solution would be to wrap.findElement call in a try-catch block and catch the StaleElementReferenceException , then we can loop and retry as many times as you need until it succeeds.