

Code can be tested by the following steps:

- 1- Import project in eclipse.
- 2- Open config.properties file and enter the file path in driver_path where chrome driver resides on your computer.
- 3- Then right click on the project and click on run as >> junit

=====

1- In case you want to check the fail scenario just enter some wrong text for example wrong subheading in config.properties and then again run the program.

=====

Project Structure

- src/main/java contains the configurations, page objects and runner packages.
- src/main/resources contains the feature package.
- src/test/java contains the seleniumgluecode package.

I have used Page Object Model (page factory)for locating elements. Every page has separate class for page object model and they all are placed under the package pageObjects.

Now my idea was to create all 6 scenarios independent of each other, so I have created one parent class known as Configure in the package Configurations. This class has all those variables which are common for every Test scenarios, not specific my Test scenarios but also the test scenarios if we want to test some other page etc. So for instance, loading the config file or setting up the browser is placed in this class.

Now in step definitions, I have used before hook which calls its parent class functions to configure settings for launching browser and loading file. Additionally javascript executor and webdriver wait is also been setup in this before hook. I have used explicit wait. Since I have used page object model, so its objects are also created in this before hook. The after hook simply closes the browser.

I have used Maven for all the dependencies.

[Chrome Version 80.0.3987.87](#)
[Chrome Driver 80.0.3987.16](#)