Read me

The following test cases has been designed to cover all the required functionality of the TODOS web page.

Testcases

|  |  |  |
| --- | --- | --- |
| Scenario | Test case | Status |
| Positive | TC01 -Create To Do task successfully | Passed |
| Positive | TC02 -Delete active To Do task successfully | Passed |
| Positive | TC03 -Mark To Do task completed successfully | Passed |
| Positive | TC04 -Clear all completed To Do task successfully | Passed |
| Positive | TC05 -Update the To Do task name successfully | Failed |
| Negative | TC06 -Add To Do task with single character name unsuccessfully | Passed |
| Negative | TC07 -Without marking complete clear completed To Do task unsuccessfully | Passed |
| Negative | TC08 -Completed To Do task comes under Active section unsuccessfully | Passed |
| Negative | TC09 -Update To Do task with single character name unsuccessfully | Failed |

What is being tested?

* UI of the web page
* Functionality
  + Positive cases:
    - Adding new task with 2 or more-character text.
    - Updating existing task
    - Deleting task
    - Marking tasks completed
    - Clear completed task
  + Negative cases:
    - Adding new task with single character text
    - Clear completed without selecting the task
    - Completed tasks doesn’t appear under active section
    - Updating existing task with single character text

Automation:  
  
How the test cases has been automated?

* The test cases are automated in Selenium using Java and Test NG framework.
* Every test case has their own class created to test the functionality.
* Locators used such as (By Id, By Classname, By Xpath) to identify the elements.
* To handle exceptions “Try and Catch” statements are used.
* Assert method is being used to set the test status Passed or Failed.

How the Test Execution & Reporting works?

* Test cases are designed to run in any possible given sequence.
* The classes are independent in nature with no dependency.
* Reporting is achieved using TestNG framework.
* After each run new report will be generated with status passed, failed and failure reason within the report.
* Additionally, a xml will be generated to run the failed test cases only.

What could be improved?

* Use of library to increase the reusability of code.
* Screenshots steps could be added.
* Use of design table in case too much of date is required.
* Integration with Jenkins to achieve CI/CD.