Introduction to DevOps

Assignment - 2022MT93076

Problem statement chosen – ***2: Automating website using Selenium.***

As part of the assignment, the website I have chosen for automation using Selenium is https://www.lumosity.com/. The automation script performs the following functions:

1. Navigate to website
2. Login to website/portal using username and password provided
3. Extract different web elements
4. Check for web elements such as tables and list the number of table web elements
5. Check for web elements such as lists and display the number of list web elements
6. Handling exception gracefully
7. Usage of locators in Selenium like Classname, ID, Tagname, CSSselector, LinkText, xpath has been demonstrated throughout the program.
   It has to be noted here that there are certain differences between css selector and xpath.
   - Xpath allows bidirectional flow which means the traversal can be both ways from parent to child and child to parent as well. Css allows only one directional flow which means the traversal is from parent to child only.
   - Xpath is slower in terms of performance and speed. Css has better performance and speed than xpath.
   - Xpath allows identification with the help of visible text appearing on screen with the help of text() function. Css does not have this feature.
   - Customized css can be created directly with the help of attributes id and class. For id, the css expression is represented by # followed by the id [#<<id expression>>. For class, the css expression is represented by . followed by the class [.<<class expression>>]. Xpath does not have any feature like this.
   - Xpath expression is represented by [//tagname[@attribute = 'value']. The css expression is repression is represented by [tagname[attribute = 'value'].
   - There are two types of xpath – absolute and relative. But css has no such types.

8. A single script is written and parallel execution is triggered on Chrome and Edge browsers thereby supporting multiple browser testing
9. thread.sleep() is used in few places to ensure that the web page is loaded completely before the program tries to interact with web elements. Thus, sleep functionality plays an important role in automated testing to ensure that the system is ready for inputs from the program.
10. We can observe from our automates that it takes about 1 minute to execute and on two browsers.
    The time would have been the same had it been 3 or 4 browsers too since the execution is parallel.

One can image the manual effort it might take to execute this on a single browser. Let us take for example that the functionalities we covered in this automation takes 3 minutes if done manually. 3 minutes on one browser and if we are considering 4 browsers then it would be 4*3=12 minutes that is huge time and effort needed. Automation addresses these constraints and helps in reducing repetitive tasks considerably.
Manual tests are time, cost, effort intensive, while automation scripts once developed, save us from time and cost and effort involved. The only effort we can probably see is to maintain is scripts in cases where it fails due to script issues.

Link to Video Recording and files on Google Drive:

https://drive.google.com/drive/folders/1HANdL6mgpipCxTrzDsTkVzEtbnWgY-W1?usp=share_link

Link to .zip file of the project can be found in GitHub link:
https://github.com/202276/DevOps-Assignment---Sem-1/upload

Files shared on One Drive:

1. SeleniumTest.java– *Main file that containing the executable code*
2. 2022MT93076-DevOps Assignment Recording – *Video recording of the assignment*
3. testing.xml *–xml file to perform cross browser testing in parallel*

Files shared on Git Hub:

1. SeleniumTest.java– *Main file that containing the executable code*
2. 2022MT93076-DevOps Assignment Recording – *Video recording of the assignment*
3. *testing.xml –xml file to perform cross browser testing in parallel*
4. *Drivers needed to execute the script*

*Please note that the drivers will need to be added if executions are done on another system and relevant changes have to be done in the file "*SeleniumTest.java*" in System.setProperty*