Homework: OneDrive Automation

Overview

Create a selenium test framework to automate the OneDrive web application. Please setup a free OneDrive account on https://onedrive.live.com. The automation should cover at minimum the following test cases:

- Login to OneDrive web application using user credentials.
- Attempt to upload a 0 byte file to the documents folder. Verify and track the status of the upload.
- Attempt to upload a text file with content to the documents folder. Verify and track the status of the upload.
- Select the uploaded document and click on the info button in the top right of the screen. Verify metadata about the document is correct by comparing it to the source file.
- Update the contents of the text document from the OneDrive editor and click on save button to create a new version of the document in OneDrive.
- Download both the versions and compare the contents to make sure they download correctly.
- Delete the document from OneDrive.

At the end generate a report to include the following following metrics:

- Number of defects found.
- Time required for testing all these test cases for each and every release cycle.
- Status of each test case.

Required Features

- **Selenium**
  - Your project must make use of Selenium for automating the browser, and work with Google Chrome.

- **Language -** *choose the one you are more comfortable with*
  - C# using the .NET framework
  - JavaScript using Node.js

- **Scalability**
  - The project should be designed with scalability in mind, for example using a page object model. Even though it is only testing a small portion of the site, it should be designed with the intention of being easily scaled to test the rest of the website.

- **Full Automation**
  - Your test suite must be able to be run multiple times, without any manual steps in between to reset.

Submission

What you should provide:

- A folder, archive, or hosted source repository (i.e. github) containing
  - Source code
  - README
  - Sample files
  - Any other files required for compiling the solution

- The README should contain:
  - All information needed to run the tests
    - If external libraries are used please provide name, version, license, and website for each.
  - A description of each test that it runs including everything that it verifies and why you chose to verify those things.
  - A basic plan of how you could scale this project in the future.

Bonus Points

- **Beyond GUI**
  - Include in your README how you might extend this test framework to make use of other types of automated testing beyond the GUI layer.
- **Additional Test Cases**
  - If you have time, add steps and verifications to this, or include other end to end scenarios.