Selenium Testing Documentation

Frameworks and Tools:

- Selenium IDE for Chrome/Firefox - https://www.selenium.dev/selenium-ide/
- Selenium WebDriver – Install through the NuGet Package Manager or Package Manager Console with command below.
    - Install-Package Selenium.WebDriver

Process:

- We used Chrome, but Selenium IDE can be installed for Firefox as well. Using the link above, download the IDE extension for your browser. After installation, open up the extensions and click on Selenium IDE.
- Click on "Record a new test in a new project" and after entering a name it will then ask for the project's base URL, ours being: https://bucstop.mjustis.com/
- After this, a new window will pop up with the website and it will say "Selenium Recording" in the corner of the window. Now, perform your test by clicking on different elements of the website. Test the navbar elements and all links on each page and submit forms for email and the "Add Your Game!" page. When finished, close the browser window and your test will be populated in the Selenium IDE window.
- Here you can edit your test by clicking on the commands and altering them if you choose to. For example, we can click on our type command for the email and change the value from "test@etsu.edu" to something else.
- Hit the save project button and save it somewhere on your machine.

Exporting Tests:

- On the left side of the IDE window, you should see your untitled test. You can hover the mouse over the test and then click on the three dots that pop up on the right side to bring up more options. Click Export and then click C# NUnit. You can select the "Include origin tracing code comments" and "Include step description as a separate comment" options if you want the test to include extra comments for each step.
- This exports a .cs class containing the test code.

Running Tests and Error Logging:

- Since our testing class is already set up all we need is the test code, we don't need the extra stuff generated by Selenium IDE. Copy all of the code within the [Test] method and paste it into the BucStopTesting Program class under the Try{} block, overwriting the previous testing code.
- In our Program.cs class, we first get the path for our error log, and then we initialize the driver for the browser. It's set up as a Chrome driver, but this can be changed to use any supported browser. The Try-Catch block is where the main test is performed. It runs through each command and catches any exceptions that might occur. It also logs each exception thrown into a text file that is located at the file path ~/Bucstop/bin/Debug/Net6.0/Error Log/log.txt