**What Went Well**

One of the positives from this sprint was the ability to lean on the provided reference material, which helped guide the creation of the test design. The setup wasn’t overly rigid or prescriptive, which I appreciated—it allowed for flexibility and supported the way I tend to learn, by picking apart different pieces to understand how they work. This approach helped me dive deeper into the use of various methods across different class objects in Selenium and Cucumber, and I now feel more confident in how to use them effectively in the future.

Although there were moments of frustration, they ultimately contributed to a better learning experience with the frameworks. Another aspect that went well was the test data I created. I believe it covered a wide range of edge cases, and when it came time to apply it, I was able to narrow it down effectively based on the different scenarios. Lastly, I was pleasantly surprised by the transition of my tests from failing to passing. I feel like my tests maintained a good balance and realistically simulated user input, which contributed to their success.

**Points of Improvement**

One area I need to improve on is developing a clear project structure before I begin writing code. I found that a lack of upfront organization hindered my progress during this sprint, as I ended up with files and code scattered in places they didn’t belong, leading to redundancy and inefficiencies.

I also want to improve the way I write acceptance criteria. I had to revise mine multiple times because some of the steps were overly detailed, which resulted in writing more steps in Cucumber than necessary. Streamlining this process will help reduce rework and make my test scenarios cleaner.

Another area for growth is my use of Jira. Since I worked on this sprint individually, it was harder to grasp Jira’s full functionality and purpose. I believe that in the upcoming team-based sprint, I’ll gain a better understanding of how to utilize it effectively to stay organized and productive.

Overall, I think I just need to get better at syncing up the planning and execution parts of the process. A lot of that will come with more hands-on experience, but I also want to be more intentional about improving how I work. I'm hoping that with more practice and especially with the team setup in the next sprint I’ll start to find a better rhythm.

**Goals in Sprint 2**

In the next sprint, one of my goals is to start incorporating “Rules” in Cucumber. We covered them toward the end of this sprint, so I didn’t have enough time to apply them meaningfully. I believe they’ll help me structure my acceptance criteria more clearly and allow me to focus on specific aspects of each user story without creating unnecessary overlap in my tests.

I also plan to integrate JUnit to improve the clarity of my assertion failures. During this sprint, I saw methods in Junit that provided detailed failure messages when running the test and wanted to incorporate them to help with my troubleshooting. Although I didn’t have the chance to fully implement this, I see the value in being able to pinpoint issues more easily without relying heavily on debugging tools or println() statements.

In addition, I’d like to refine how I use Jira, particularly for defect tracking. Right now, the interface still feels a bit complex, especially when working solo. However, I’m confident that with more experience and especially in team setting I’ll be able to focus on the features that are most relevant to my workflow. Shifting away from informal documentation and using Jira more effectively will be an important step in mastering the Scrum workflow.