Assignment 5

Reethu Chowdary Vattikunta

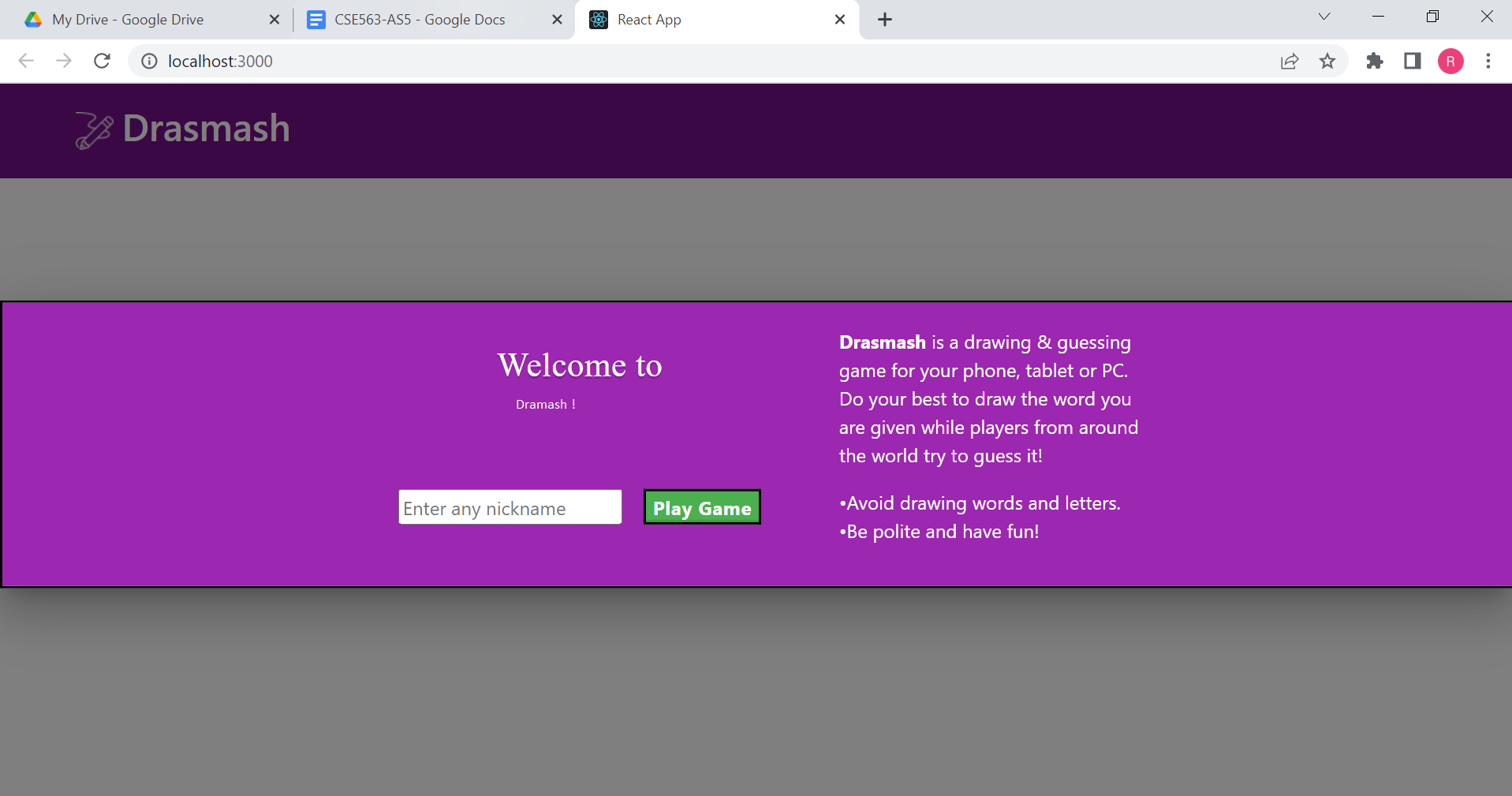
1222619619

**Description of the application:**

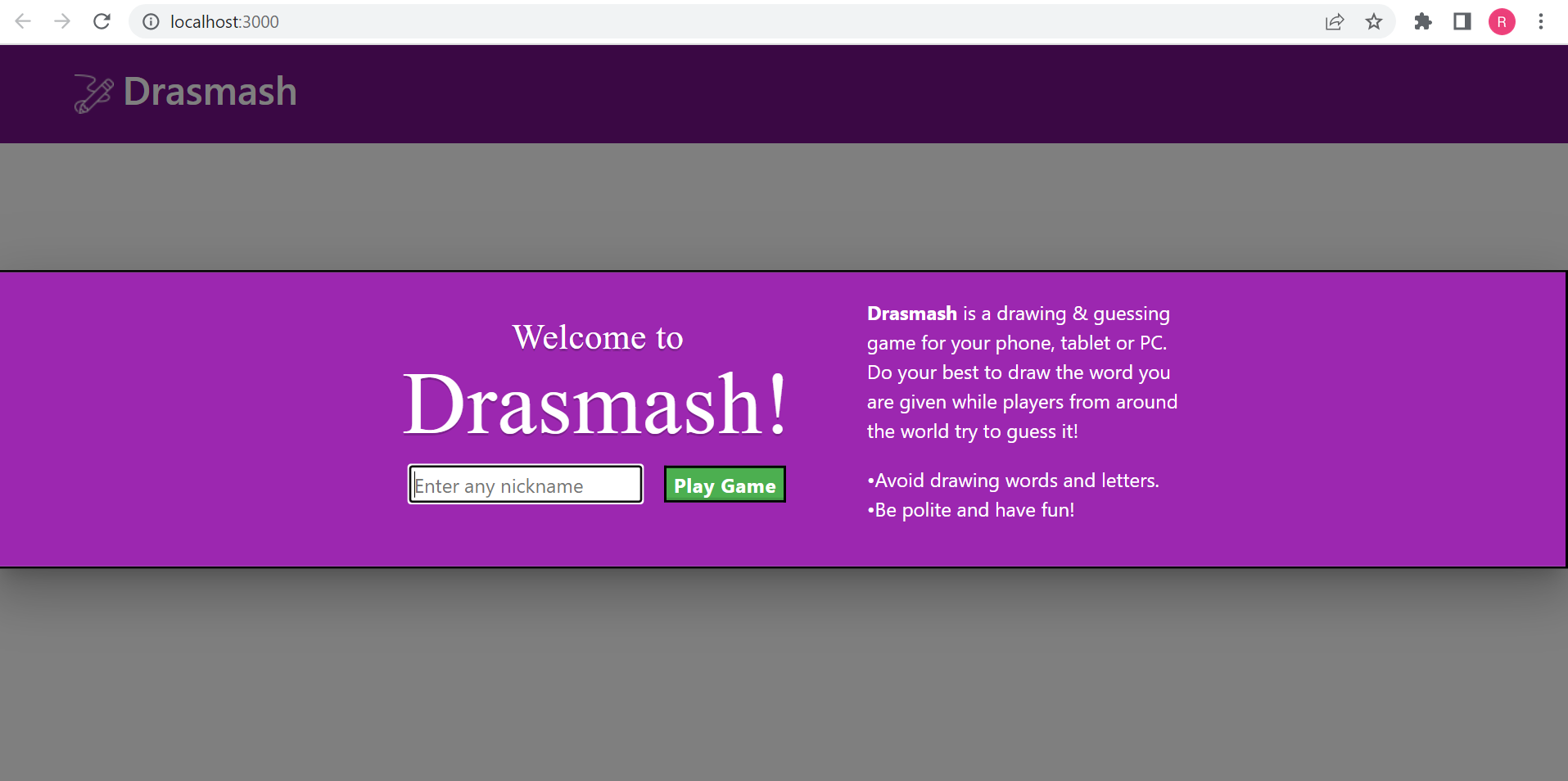
This open source website is very much similar to the famous game named Scribble[[1]](#footnote-0). This accepts the user's login and lets the user create the room for other players to join. It's a very straightforward guessing game where you can select one of three ideas when it's your turn. To draw your cue and give the other participants a chance to guess. To win, you must be the fastest, most accurate, and have the highest score. You can only draw a clue; you cannot draw the solution when others are drawing. The multiplayer mode makes each round of the game unique. You will not be familiar with each person's drawing style, and it will be difficult to predict their ideas. Each round is pleasant and unexpected. The following screenshots show the GUI of the three pages.

**Version 1 vs Version 2:**

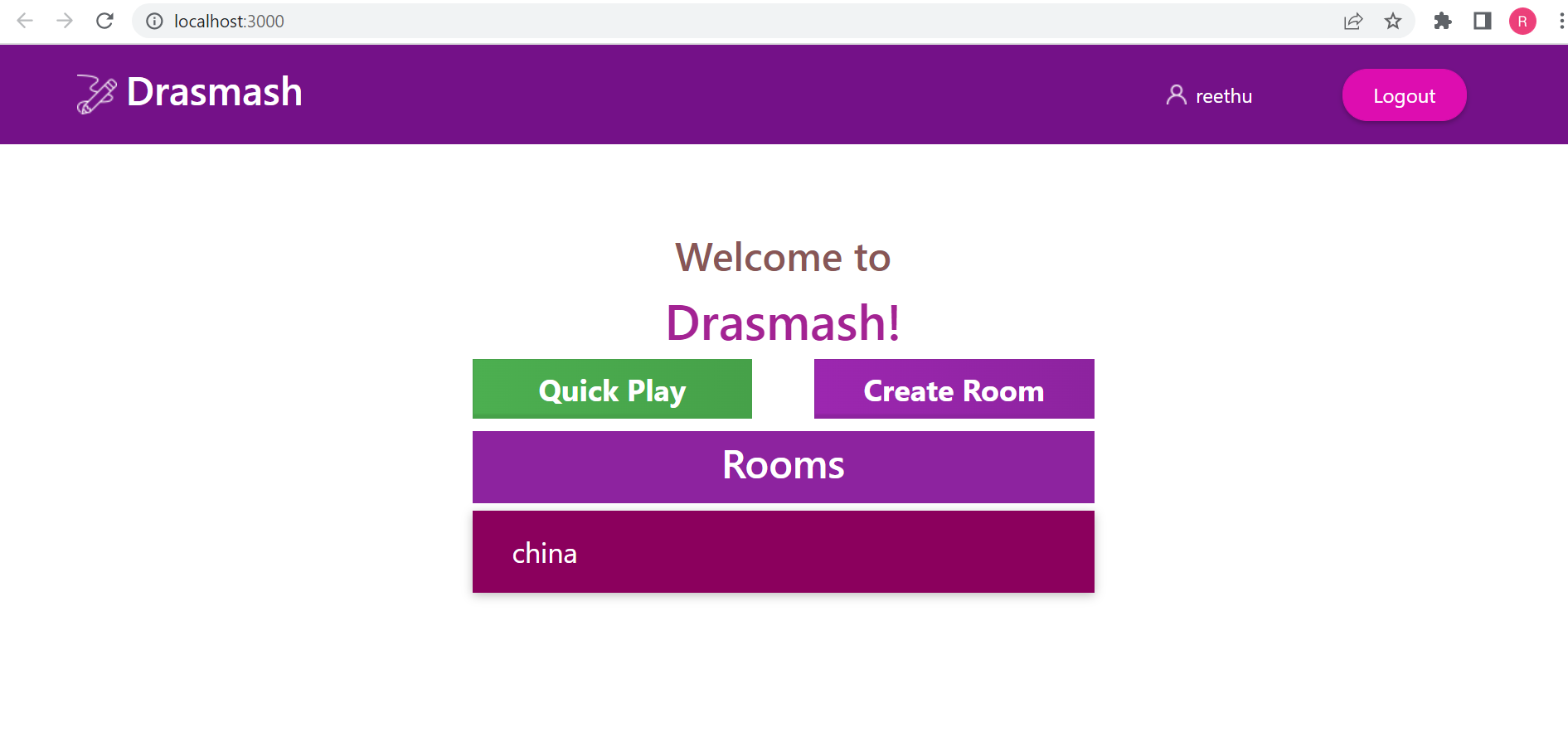
To run this web application, one has to go to the src of the client in order to do the front end testing. After opening the command prompt at the src, run “npm start”. This will run the web application and open the local host link “<http://localhost:3000/>”. Once the changes are made, rerun this link to see the changes. The page will reload as soon as you make changes. Grain errors may also be displayed in the console. Although there are few changes from the version 1 and version 2, version 2 is made sure to have better flow and user usability. The font changes are seen in the first GUI page. The entire layout of the rooms page was changed by changing the sizes and locations of room buttons on the page, as well as text sizes. The flow steps are made sure to move from one page to the next without any hustle. All the flow steps shown clearly with both the versions starting from login to playing the game and logging out.



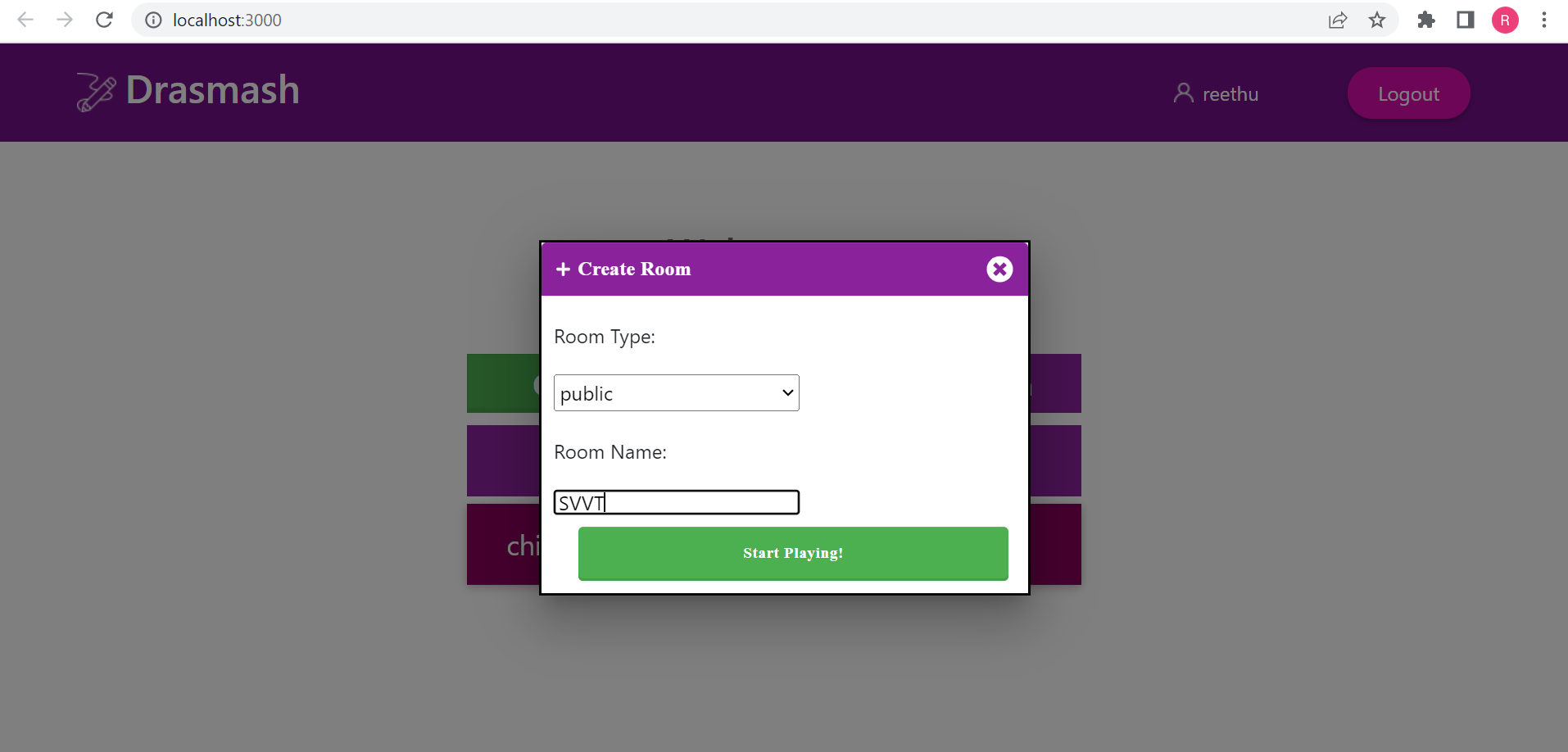
Version 1 : GUI page 1- To login with username



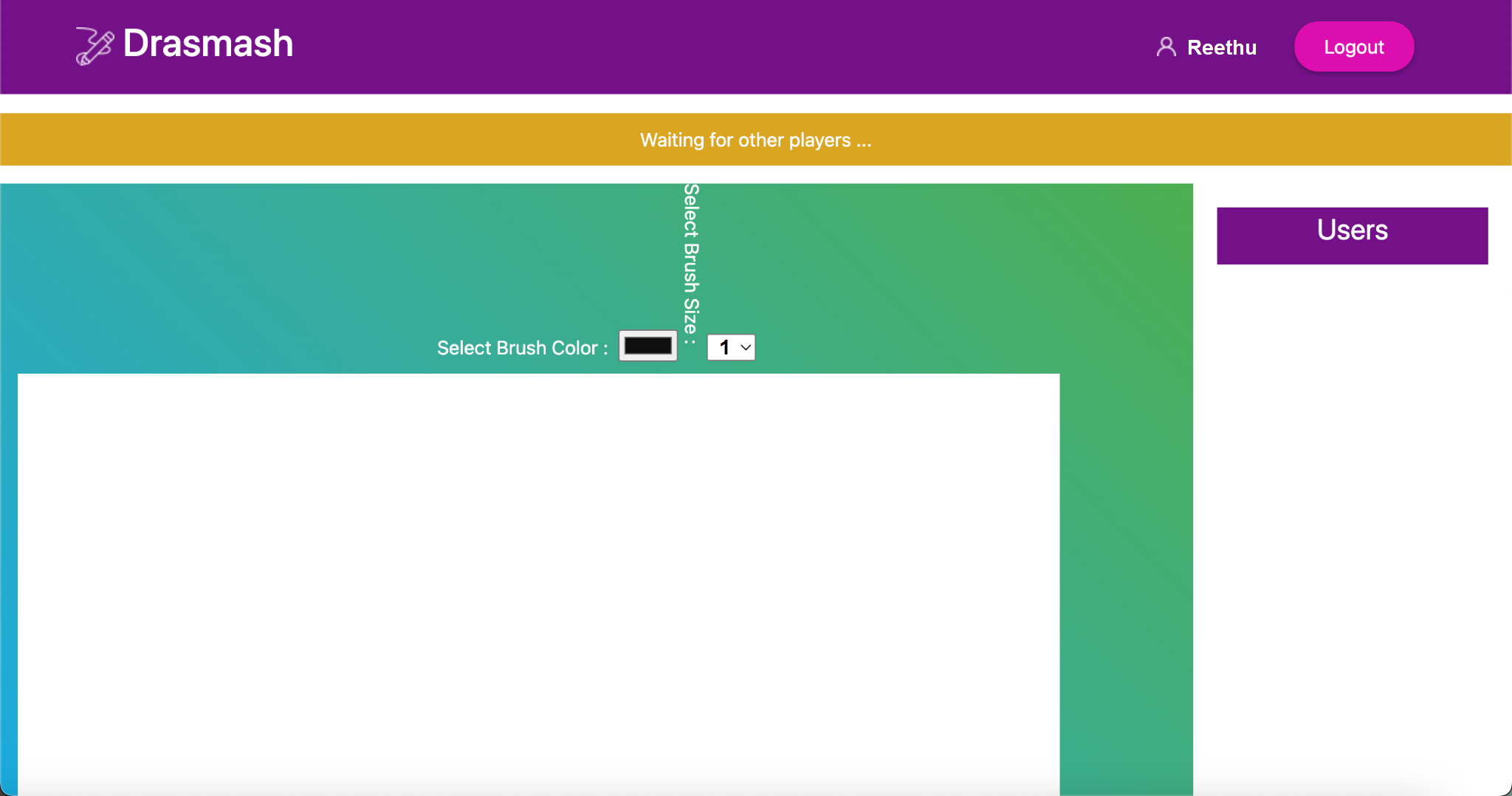
Version 2 : GUI page 1- To login with username



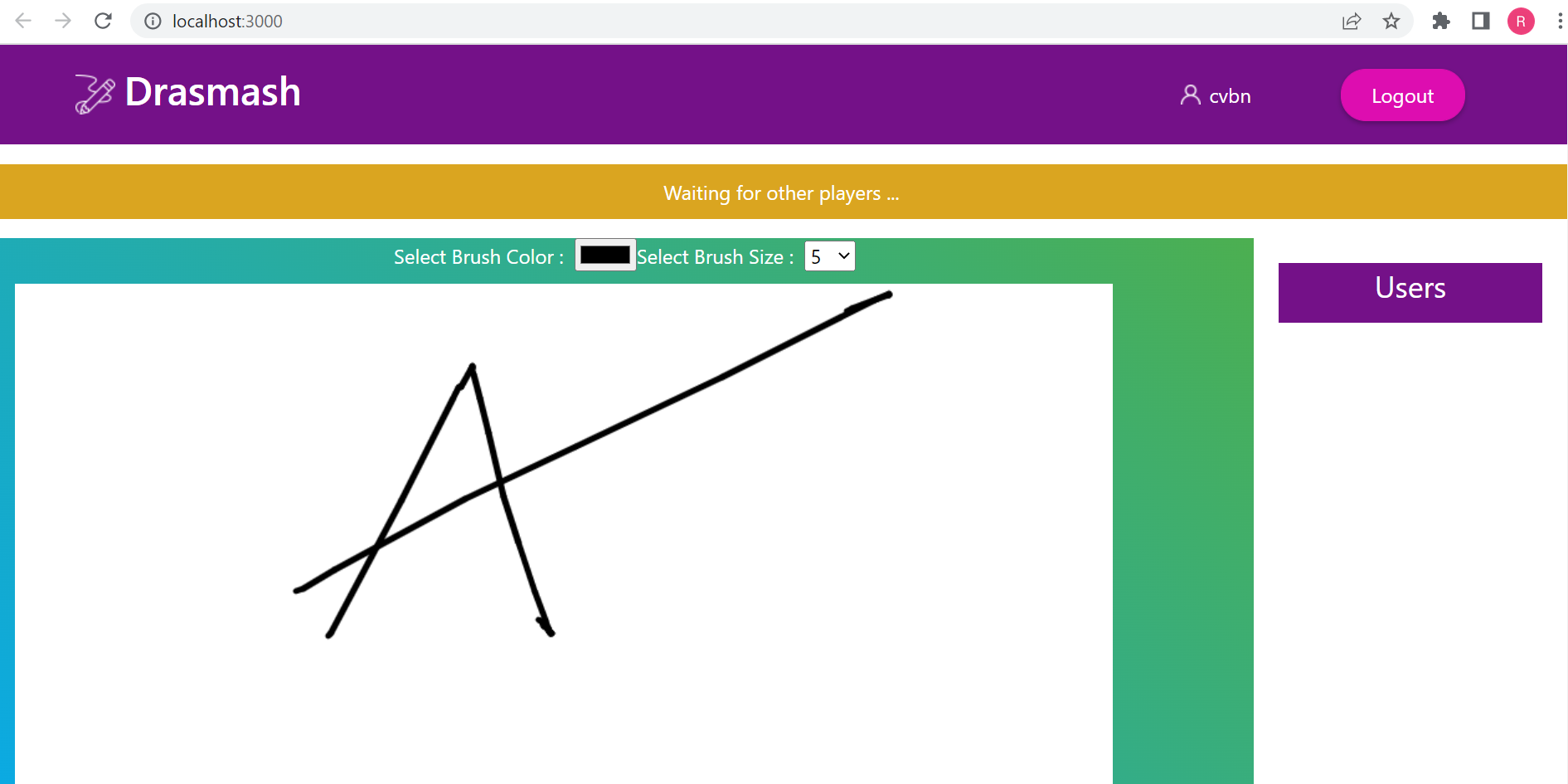
Version 1, 2 : Page 2 - to create a room or join previously existing rooms



Version 1, 2 : Pop page to create a room



Version 1 : Final page to draw the diagrams to guess the word



Version 2 : Final page to draw the diagrams to guess the word

**Description of the tool selected for GUI testing :**

The tool used for GUI testing is Selenium IDE. This is one of the most used record-&-playback tools by developers for web automation. This tool allows us to create test suites faster and export the test cases to different languages. We cover almost all of the features of these tools and explain the points by creating two sample test cases that will assist you in getting started with this tool for automation. Since this is an open-source tool, users can customize it to continue improving its features and get the most out of this automation platform.

The key features and functionalities are :

* **Web-ready**

A simple chrome extension of this selenium IDE[[2]](#footnote-1) is innovative solution which generates reliable tests from end-to-end

* **Easy-Debugging**

Even when tests are generated automatically, it is also easier to debug when a test case went wrong. One can modify the tests manually as well to add key features like setting breakpoints and pausing when exceptions are raised.

* **Cross-Browser-Execution**

In this assignment as we have two versions to be tested, one can even make it possible to run in two different environments which makes it easier to test versions in parallel.

* **Test case reuse**

One might need similar tests but need to make only few changes, that is when reusing the test cases are useful, the available options are duplicate, export.

Cons :

* The cons would include, there are few features that can not be verified using this Selenium IDE like the font size. Although it was able to test the box sizes, nothing can be done with the size of the font.
* It’s firefox plugin is limited and can cause incompatibility when used
* It also does not support the test script grouping

Considering the pros of this IDE, assignment is done using Selenium[[3]](#footnote-2)

**Description of the test cases developed :**



The test cases are developed to evaluate the location, existence, content of the GUI elements and hence the correctness of the flow is also verified with the test cases[[4]](#footnote-3) from page to page until logout.

The main test cases[[5]](#footnote-4) developed are:

* **Tests based on the size**

As this selenium IDE provides on to test the box sizes and frame sizes. I measures the window size of the web application when opened

* **Test based on existence**

For suppose when a user logged in, the user name should be displayed on the next GUI page. Thus proving the existence of the user name. There is also one more test which is room creation, where it’s name should be displayed when a user logged in and created a room

* **Tests based on Content**

Verifying whether the proper content is displayed whether it is a button or an input text box or just a display text. All the possible cases are verified.

* **Tests based on Location**

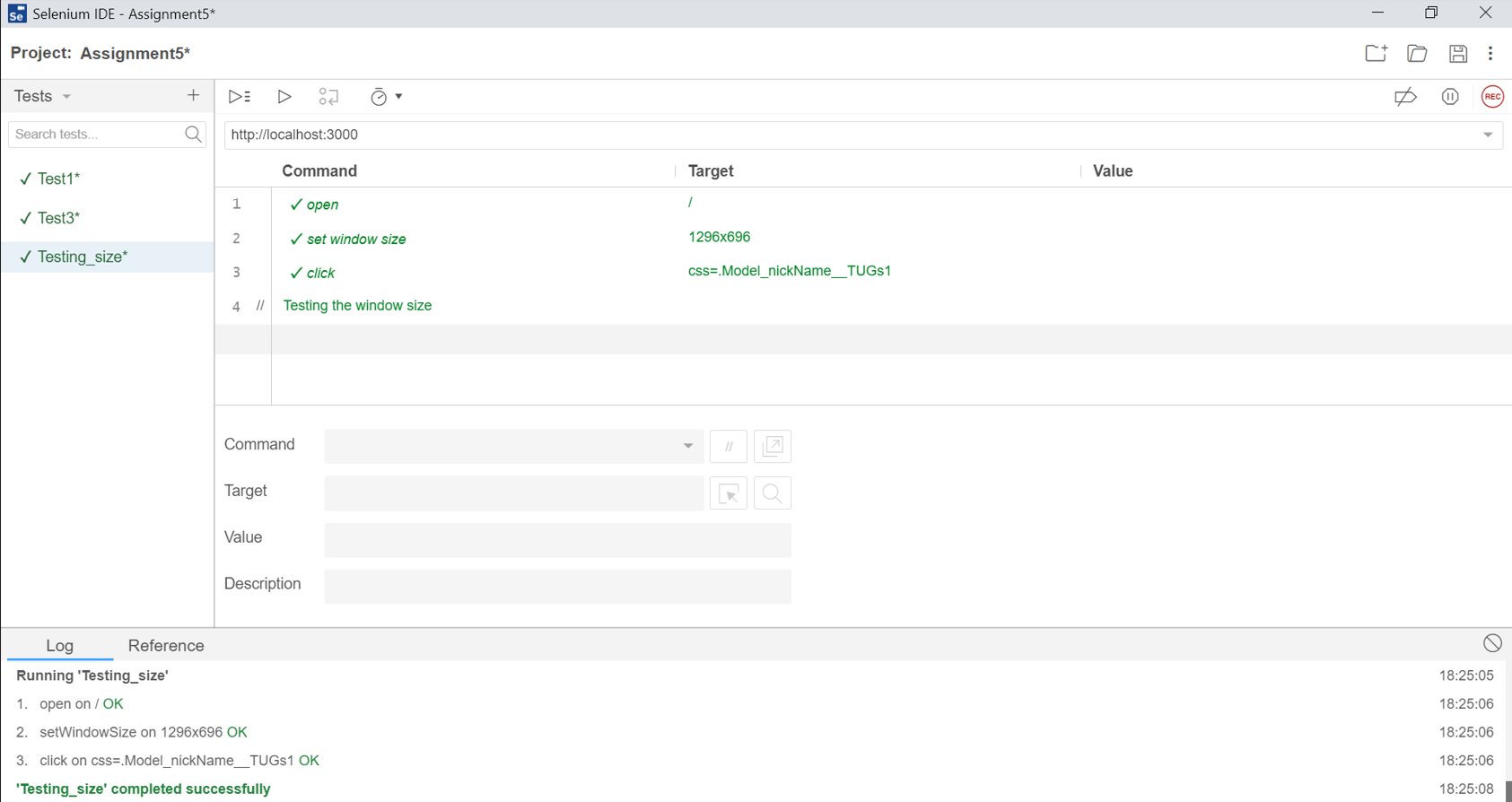
As this web game is all about drawing the images, where is it vital to have proper locations and proper movements when a cursor is moved. Tests are developed that also check whether up and down cursors moved properly.

* **Tests based on Flow**

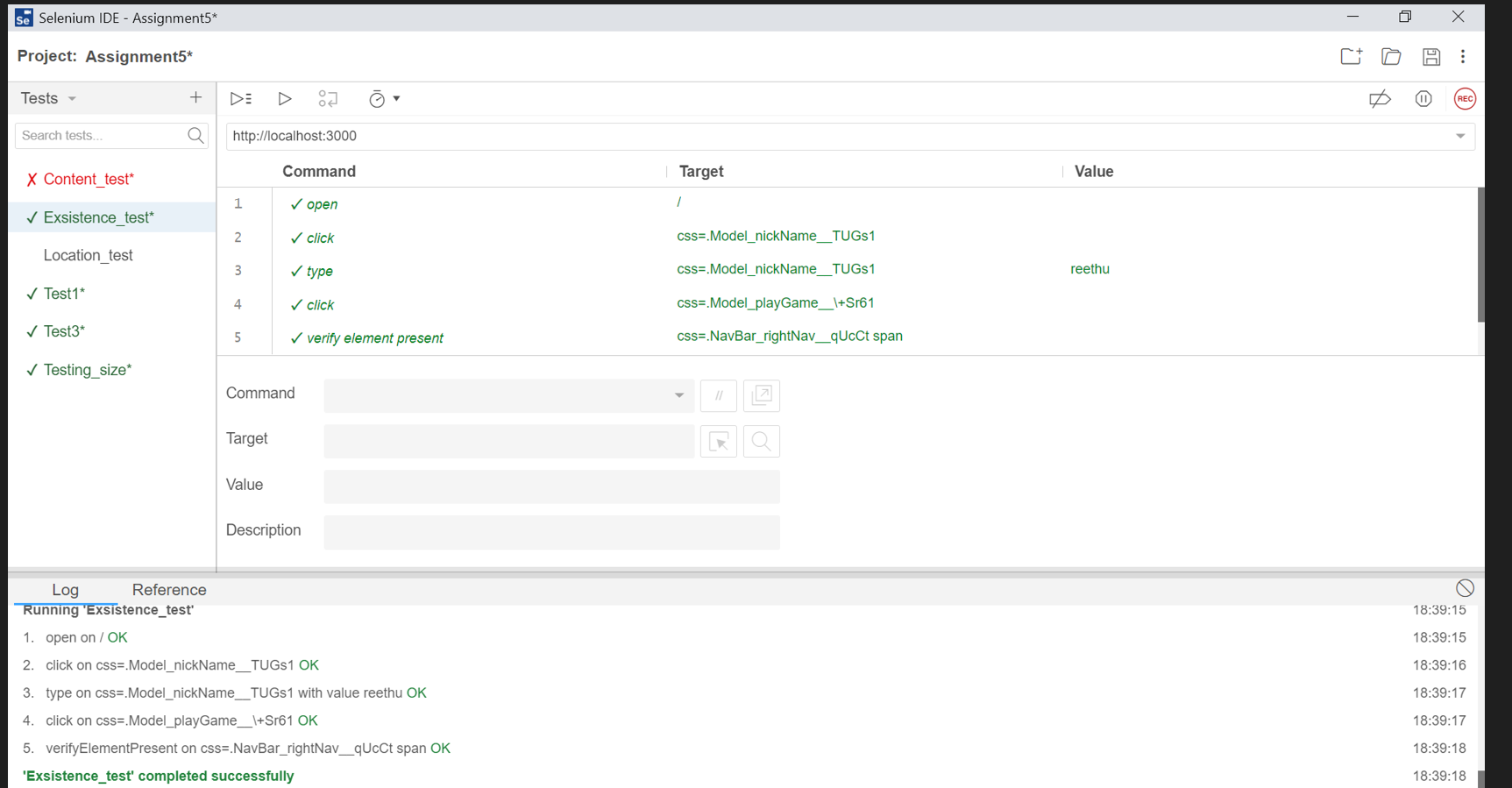
Flow between pages is vital too. From log in until log out, everything is made sure to be smooth. Testing of the flow is done in three parts between one page to another and is consolidated in one test.

***Screenshots of the Test results are displayed as follows:***

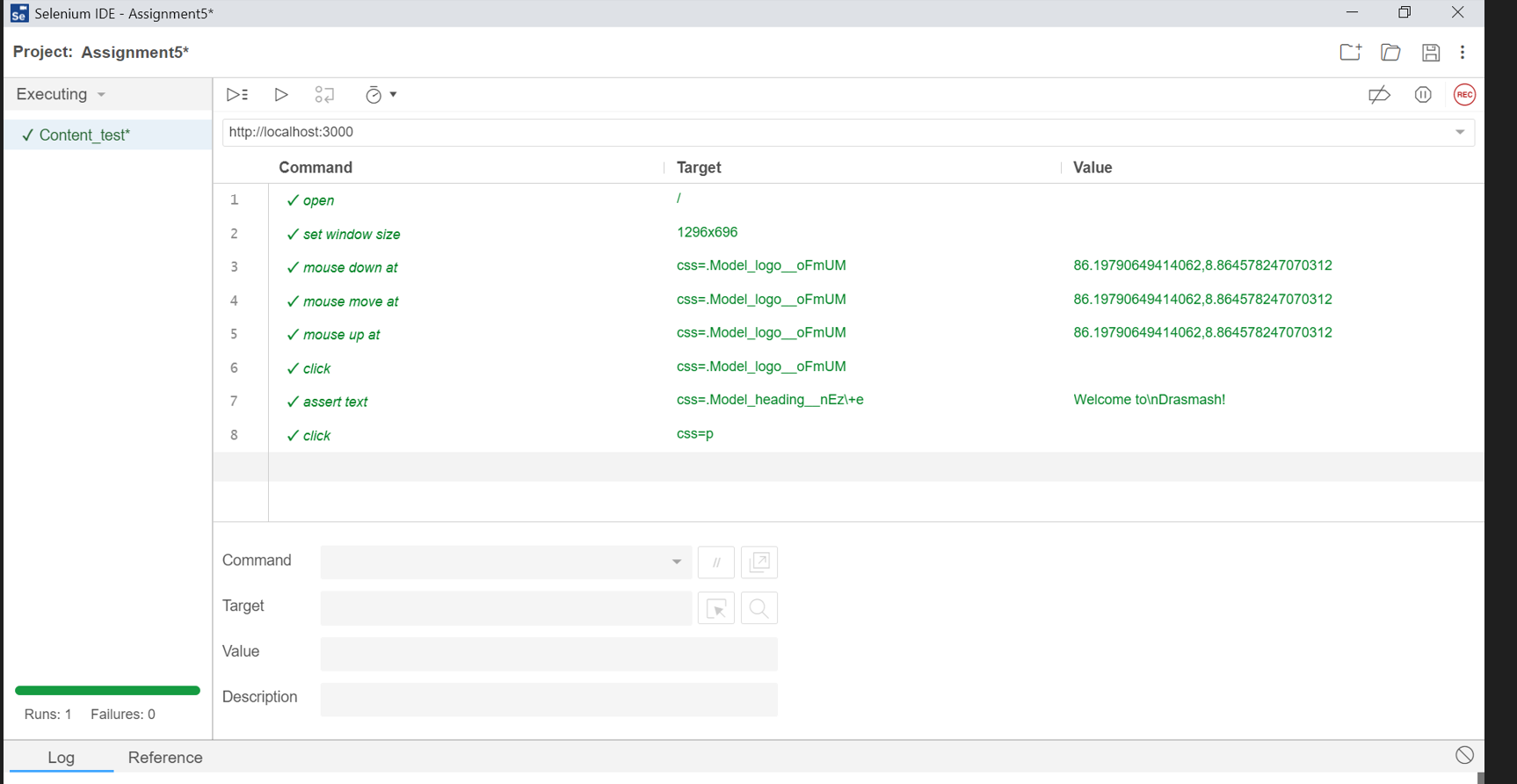
**VERSION 1 TEST RESULTS**



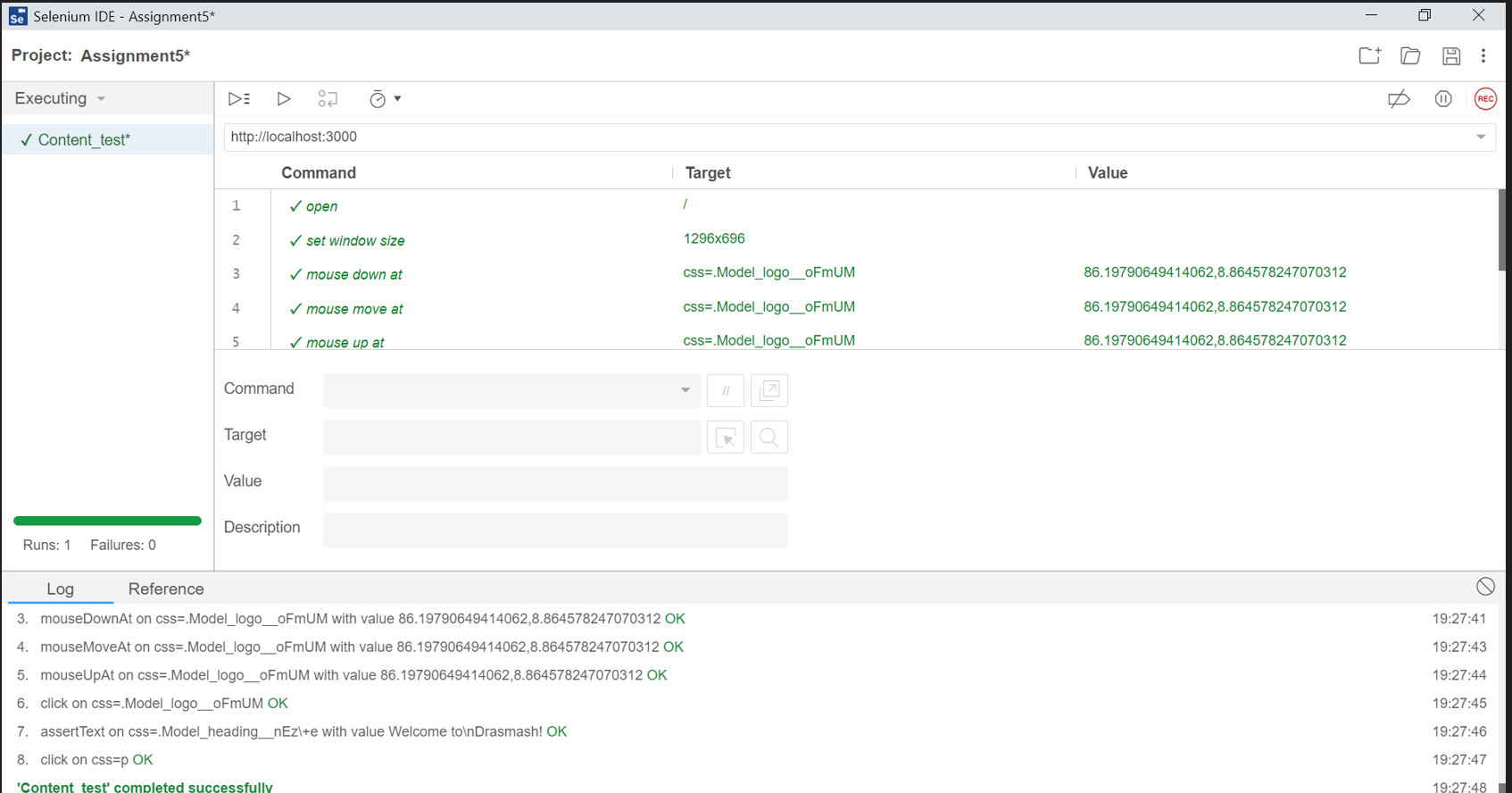
**VERSION 1 :** Test based on window size



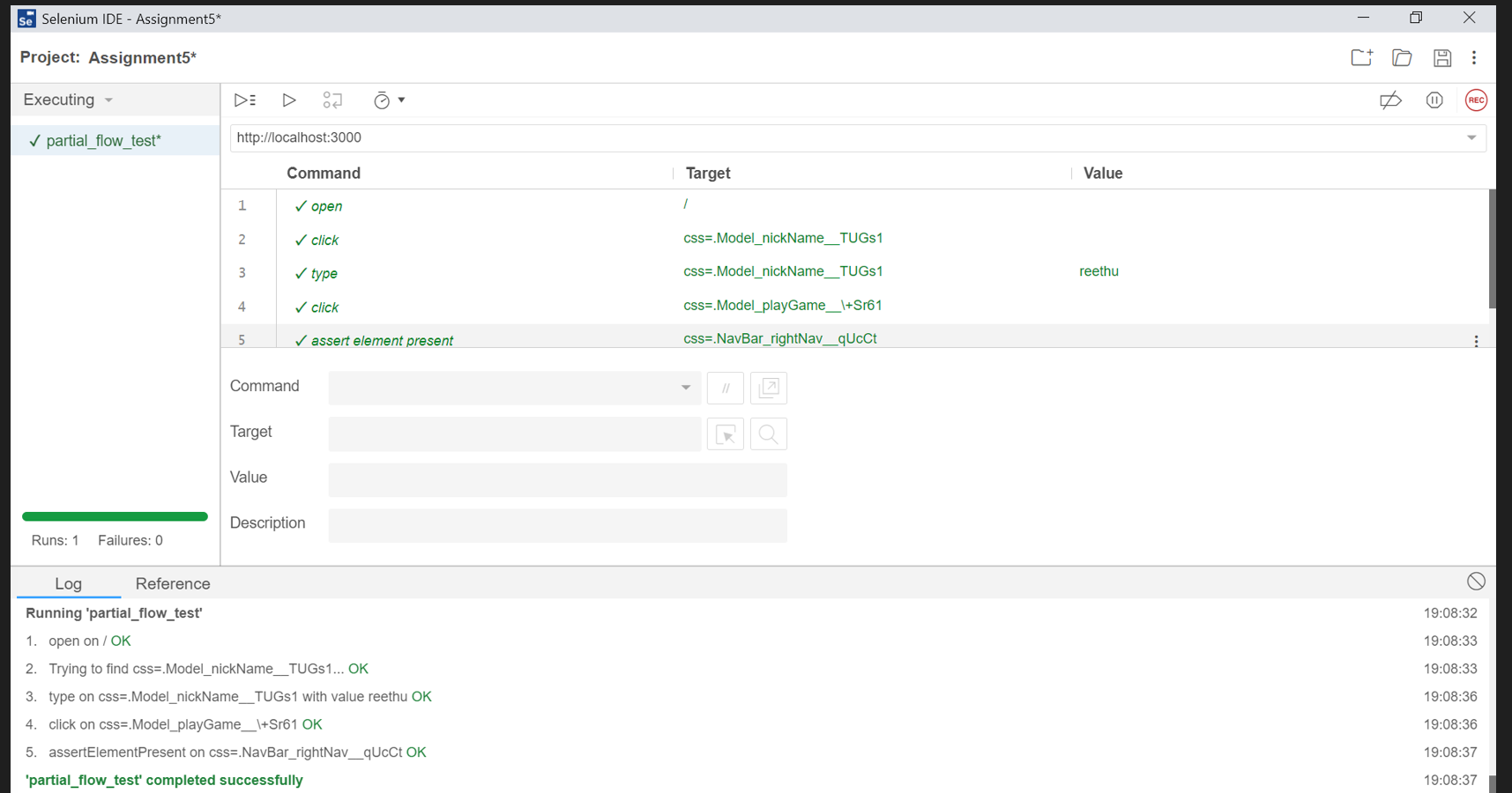
**VERSION 1 :** Test is based on the existence test



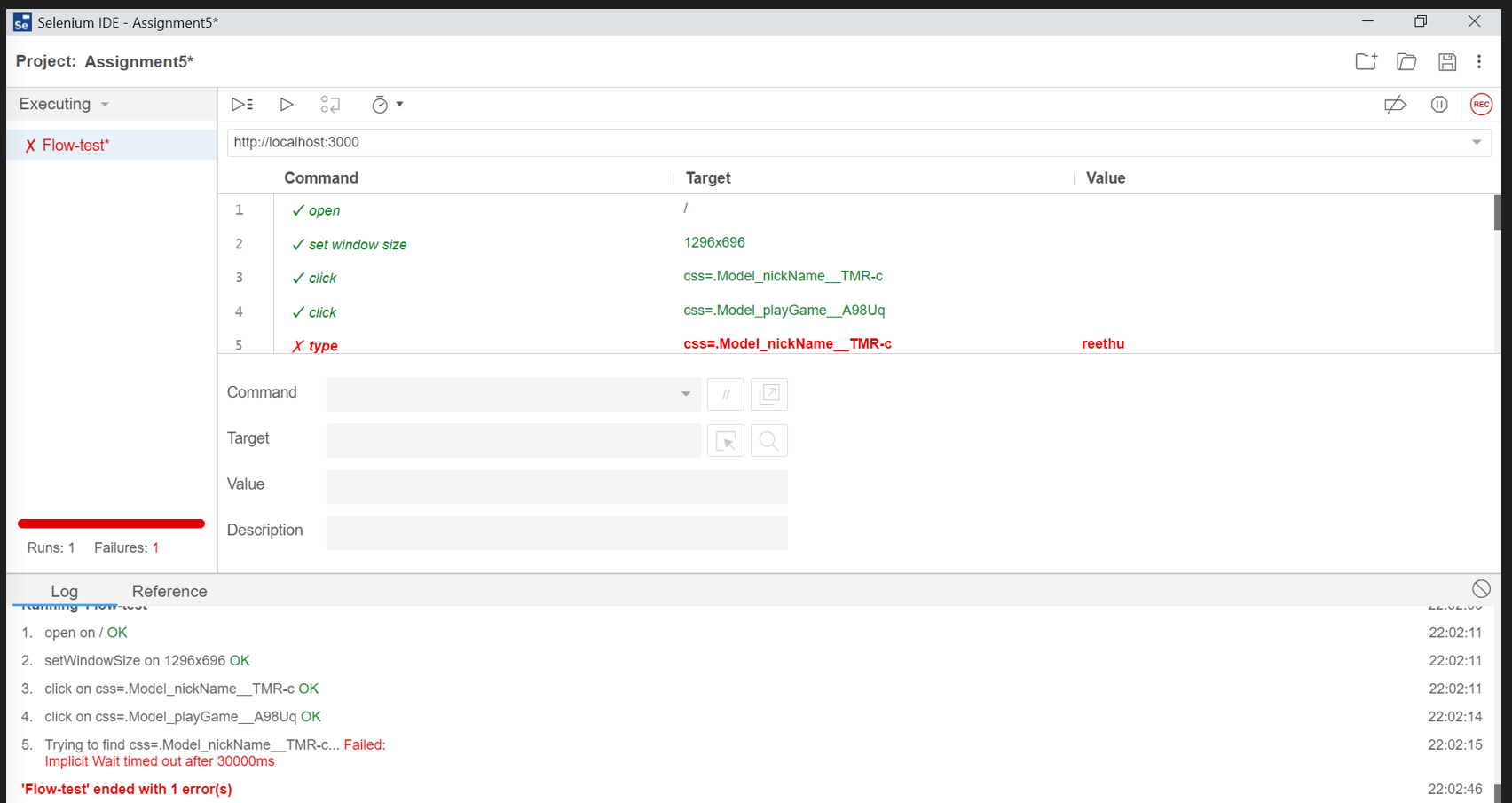
**VERSION 1 :** Test is based on the content test 1



**VERSION 1 :** Test is based on the content test 2

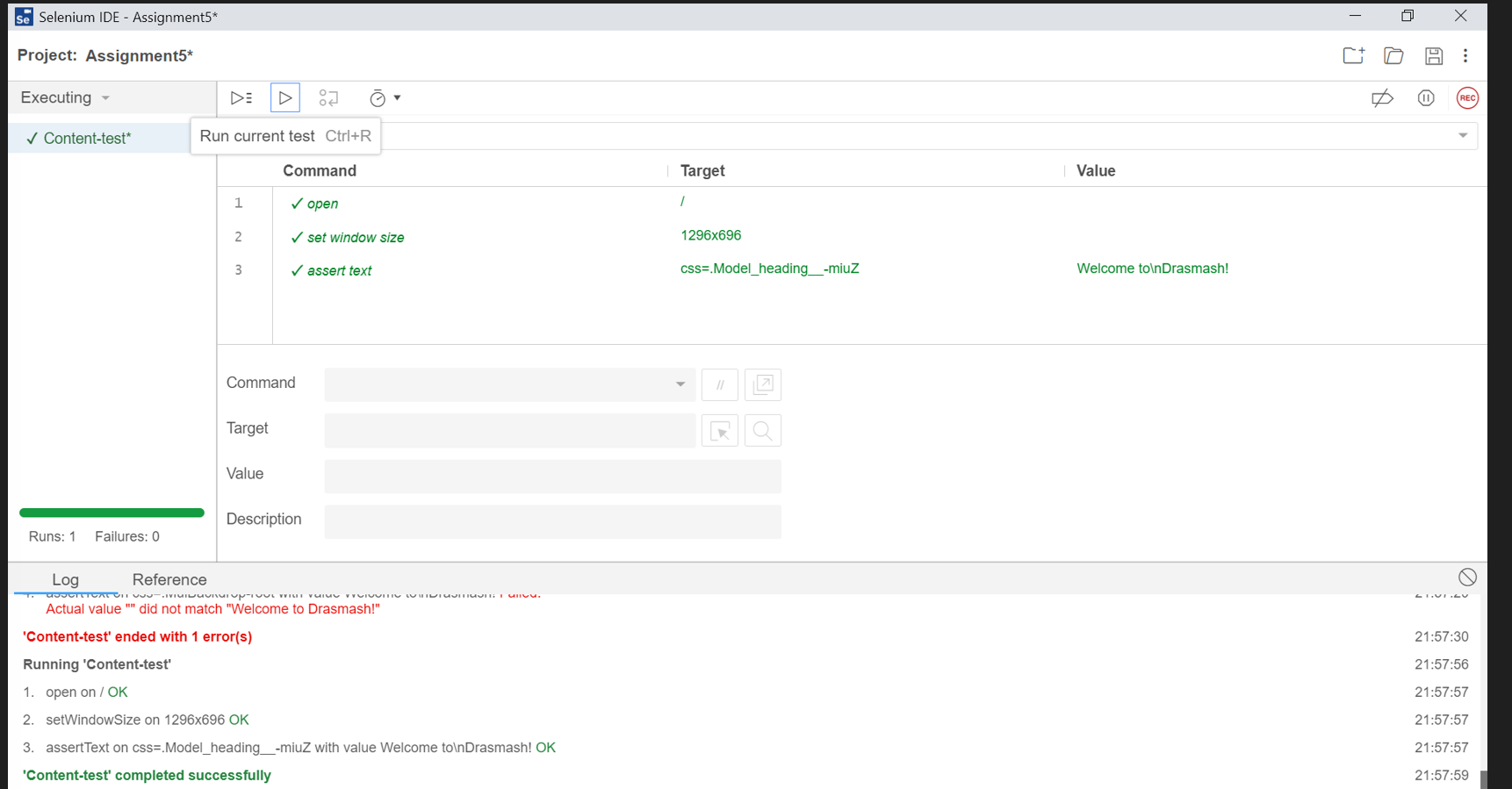


**VERSION 1 :** Test is based on the partial flow test 2 (from page 1 to page 2)

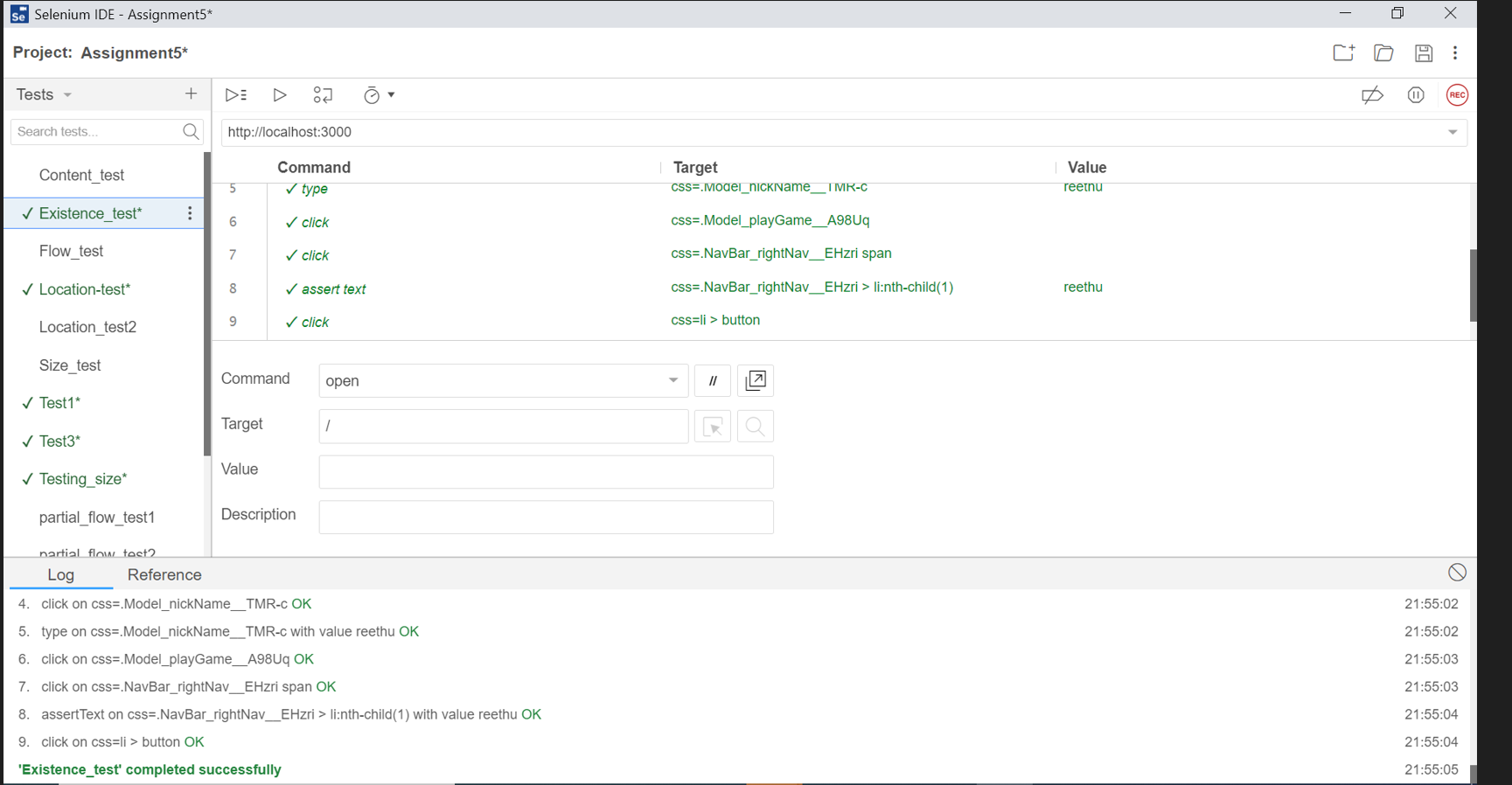


**VERSION 1 :** Test is based on the flow test 2 (this is verified in the second version)

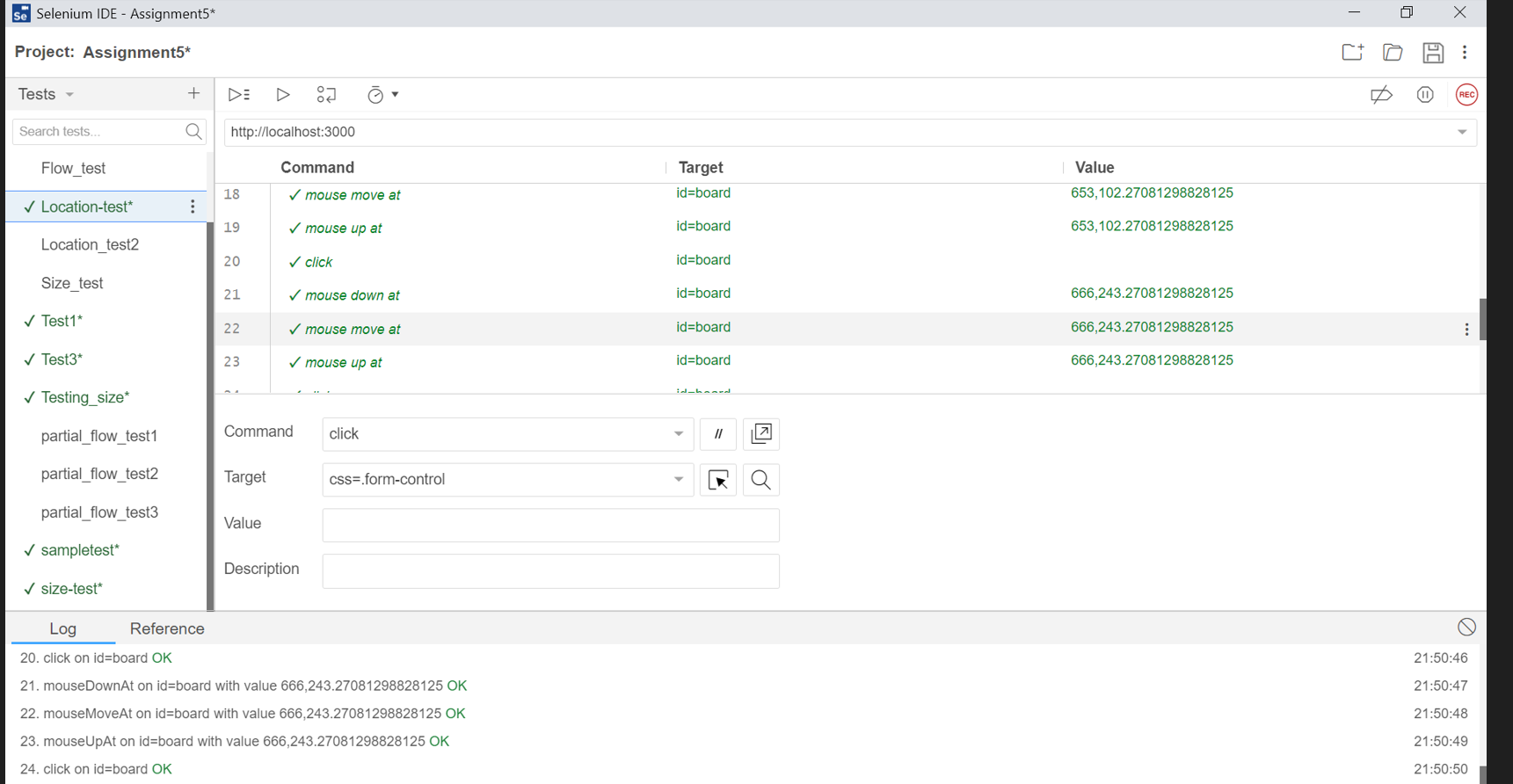
**VERSION 2 TEST RESULTS**



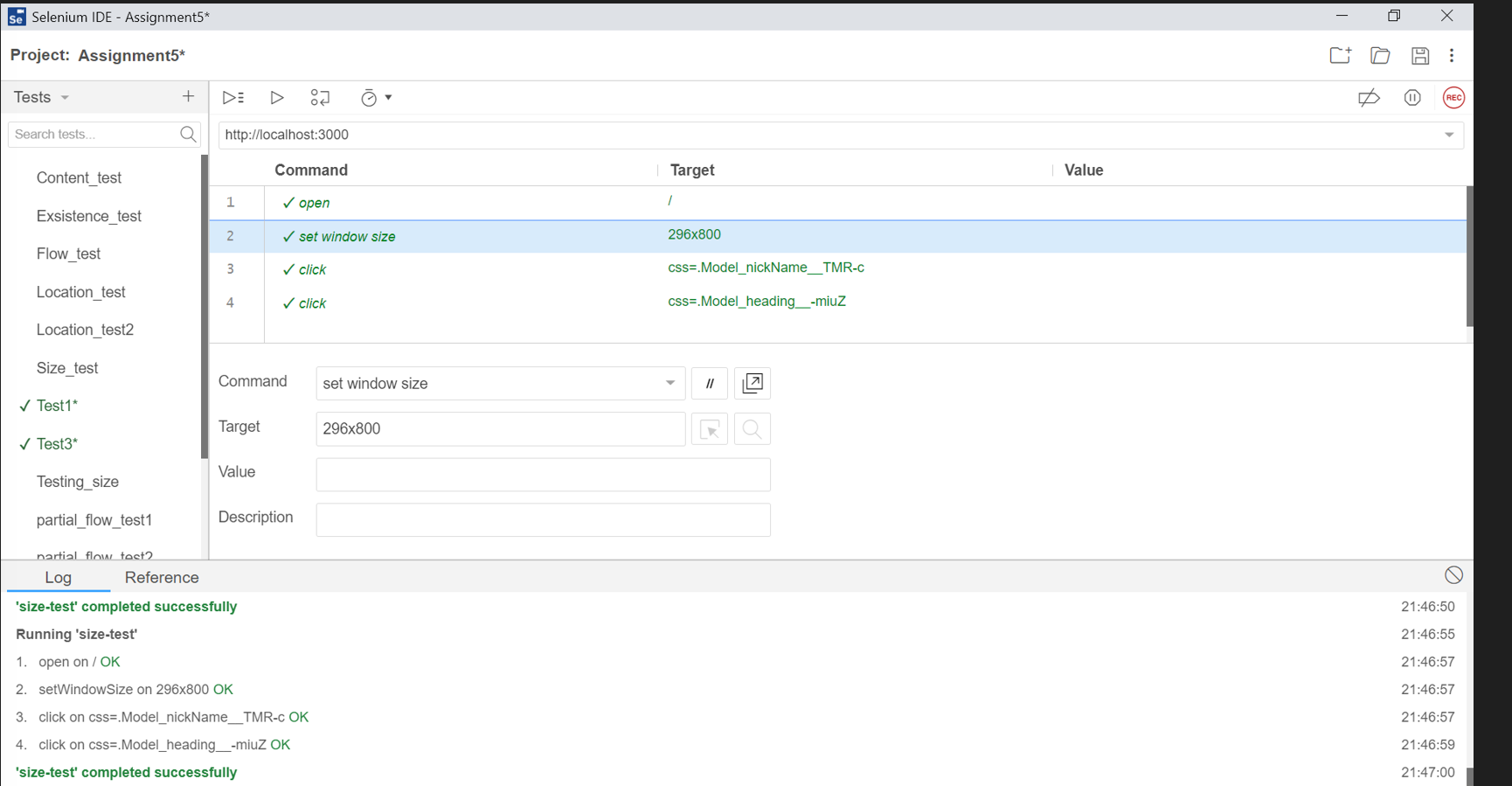
**VERSION 2 :** Test is based on the content test



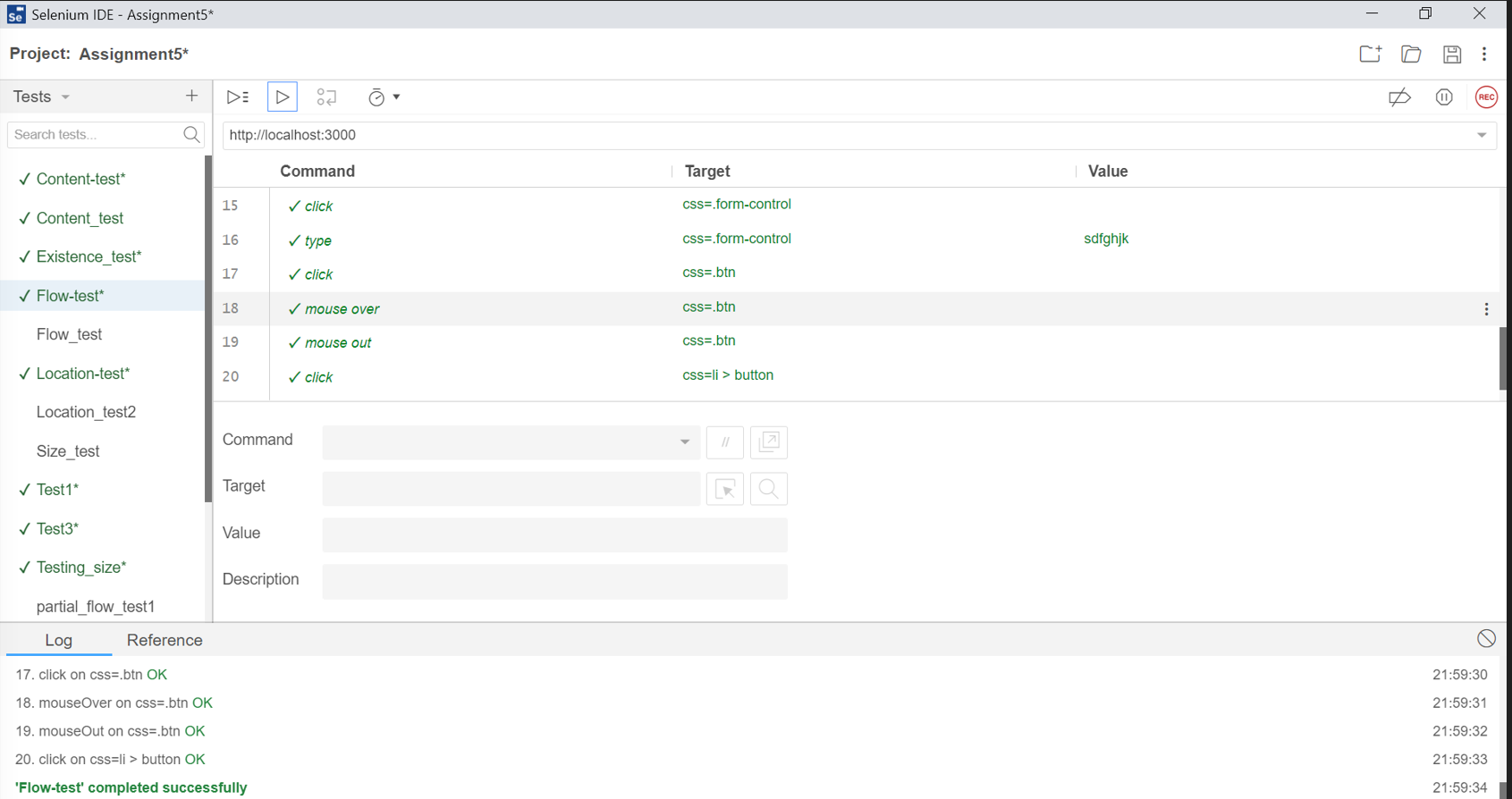
**VERSION 2 :** Test is based on the existence test



**VERSION 2 :** Test based on location 1



**VERSION 2 :** Test based on location 2



**VERSION 2 :** Test is based on the flow between GUIs

**Assessment of the tool**

Why not other tools?

I installed cypress, testComplete before using selenium, it was difficult to run cypress on chrome extension. Even after clearing security issues with chrome, there was a hard time running tests with cypress. The major drawbacks of these IDEs is that there is no proper documentation and references. The existing references are outdated and the tools are completely different from those.

Why a selenium IDE tool?

After one video tutorial on it, it was installed in seconds and ran the tests in no time. It would be best If I first started with Selenium. This tool enables us to create test suites more quickly and export test cases to different languages. We cover almost all of the features of these tools and explain the points with two sample test cases to help you get started with this automation tool. Because this is an open-source tool, users can modify it to improve its features and make the most of this automation platform. **Editing or changing the tests is never this simple.** Even when tests are generated automatically, debugging when a test case fails is simplified. Manually modifying the tests to add key features such as setting breakpoints and pausing when exceptions are raised is also possible.

1. "skribbl - Free Multiplayer Drawing & Guessing Game." <https://skribbl.io/>. Accessed 20 Nov. 2022. [↑](#footnote-ref-0)
2. "Selenium IDE." 7 Dec. 2021, <https://chrome.google.com/webstore/detail/selenium-ide/mooikfkahbdckldjjndioackbalphokd?hl=en>. Accessed 20 Nov. 2022. [↑](#footnote-ref-1)
3. "Selenium IDE · Open source record and playback test automation for ...." <https://www.selenium.dev/selenium-ide/>. Accessed 20 Nov. 2022. [↑](#footnote-ref-2)
4. "A Study on Functioning of Selenium Automation Testing Structure." 5 Jul. 2022, <https://www.researchgate.net/publication/318930970_A_Study_on_Functioning_of_Selenium_Automation_Testing_Structure>. Accessed 20 Nov. 2022. [↑](#footnote-ref-3)
5. "Selenium IDE Tutorial For Beginners - YouTube." 4 Jul. 2019, <https://www.youtube.com/watch?v=m4KpTvEz3vg>. Accessed 20 Nov. 2022. [↑](#footnote-ref-4)