CSE 565 SVVT Assignment 5: Graphical User Interface Testing

Siddharth Gianchandani

Arizona State University

Description of application with screenshots for GUI:

I have developed an open-source web application with a graphical user interface. This application was developed using HTML, CSS and JavaScript. It consists of three pages where the first page is the login page, the second page is for managing tasks, and the final page is for making changes in individual tasks. There is a data flow between the pages in the order that they have been listed, for example a person can go to the second page after they provide all the details in the login page.

Page for login:

This page consists of a single form asking for information about the individual such as name, email address, username, gender, what field they are specialized in, and a check box to remember their information for the future. This page contains a single image in addition to labels, 2 buttons, 3 radio buttons, 1 dropdown, 3 text boxes, and 1 check boxes, which comes to a total of 11 graphical elements excluding the labels.

Page for tasks:

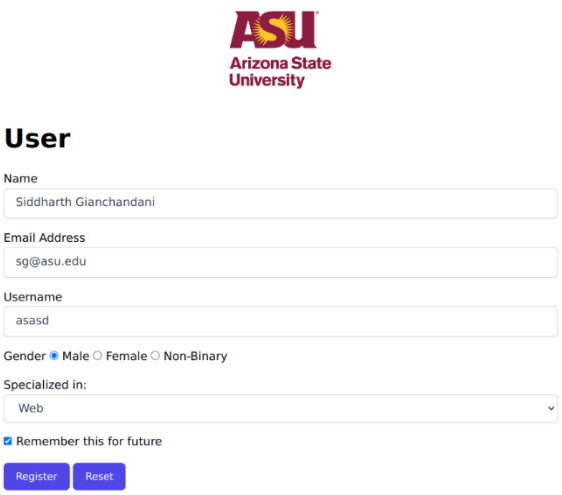
This page consists of a single form asking for information about new tasks and managing old ones. It contains a number of GUI elements in addition to labels, consisting of buttons, radio buttons, list, text boxes, and check boxes. There are three buttons for each task added.

Page for enrollment in courses:

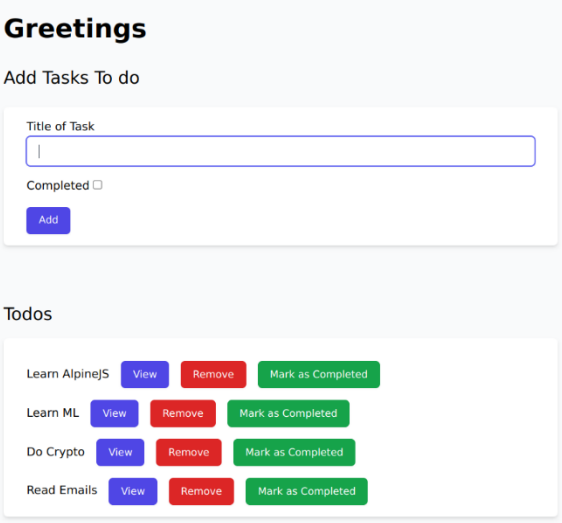
This page consists of a single form asking for information about the task such as the title, whether it is completed, whether it is urgent or not and a text area for notes. It has a total of 5 graphical elements in addition to labels, consisting of 1 text area, 1 check box, 1 button and 2 radio buttons.

Screenshots for version 1:

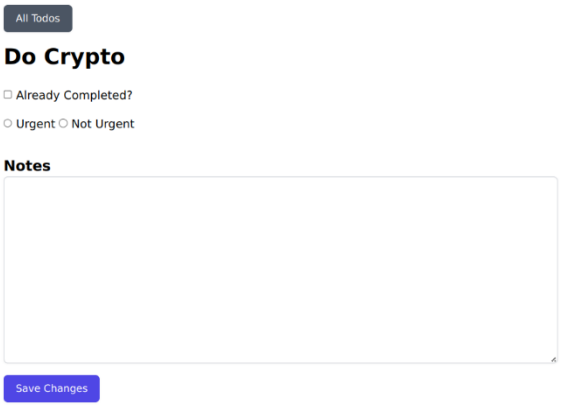
First page:



Second page:

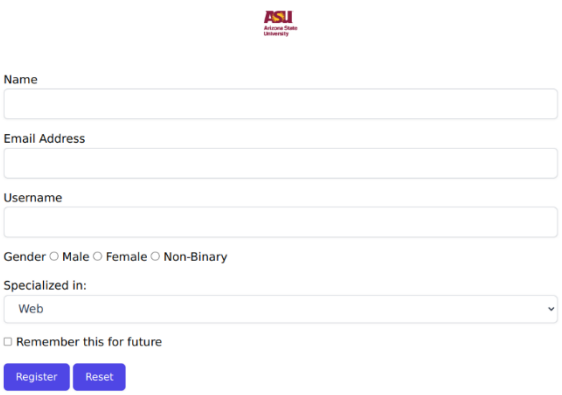


Third page:

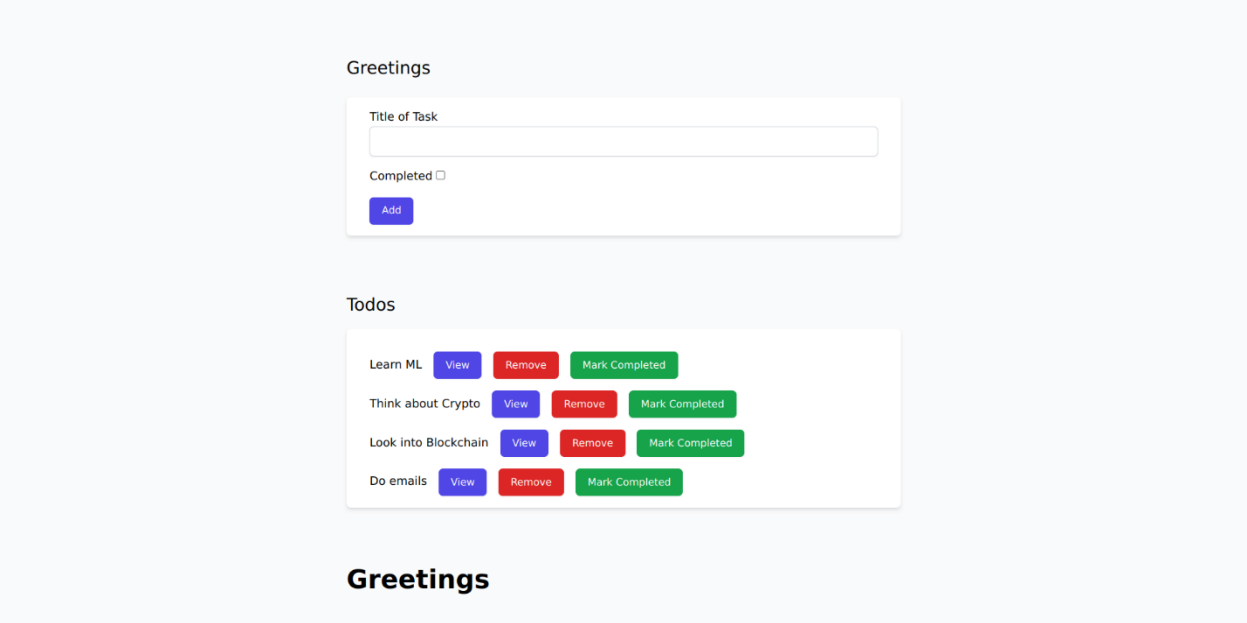


Screenshots for version 2:

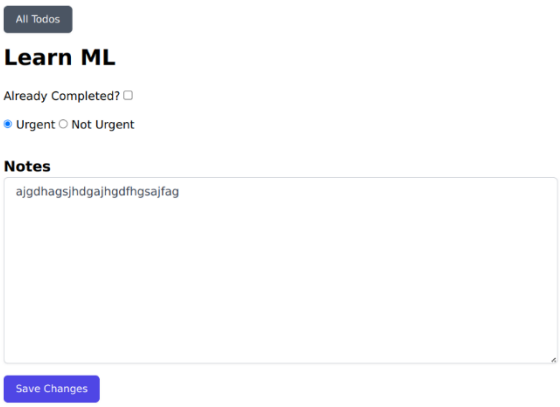
First page:



Second page:



Third page:



Description of tool:

[1] I am using the open-source Selenium IDE tool for automated graphical user interface testing. It is very useful for automating GUI test cases, especially across different versions of software. This tool is a record/run tool, and it is available as an extension to web browsers like Google Chrome without any initial setup. Since it does not require any initial setup, it helps prevent versioning issues with the project as a whole. Additionally, Selenium IDE provides a GUI with

functionalities to capture and replay the testing activities in addition to automating the graphical user interface testing, making reusable test cases. This feature maintains a record of all the GUI interactions that are done during the initial recording for each test case including inputting information into fields and clicking buttons. This feature allows anyone to write automated reusable test cases without having the required background knowledge, which has made this a very popular tool. It also does not require a tester to have all the requirements for the testing initially and allows for new test cases to be added dynamically. Since Selenium is recording the interactions between the graphical user interface and the tester, it guarantees complete coverage of all graphical elements on the pages visited by the tester during a test. All these features result in reduced requirements of time and effort needed for testing, and a reduced risk of defects being developed in the code, which makes it a great tool for automated graphical user interface testing. Finally, the fact that Selenium IDE supports multiple operating systems, browsers and languages including Windows, Linux, macOS, Google Chrome, Java and Python makes it a very desirable tool to use for graphical user interface testing.

Description of test cases:

I have generated 18 test cases for version 1 and reused those test cases to test for version 2 of my web application. These test cases were developed to test the existence of GUI elements, their location correctness, size, content and correctness of the links in each page, as per the requirements listed in the assignment description. I have listed all the changes between each version below along with the category of test cases they belong to, followed by the screenshots for the test cases.

Change to test existence of element:

I have hidden the element “User” in the second version of the software.

Change to test for location correctness:

For version 1, the greetings label was at the top of the second page, while it was at the bottom for the second page.

Change to test for size:

I have made the size of the image smaller for the second version of the software.

Change to test for content:

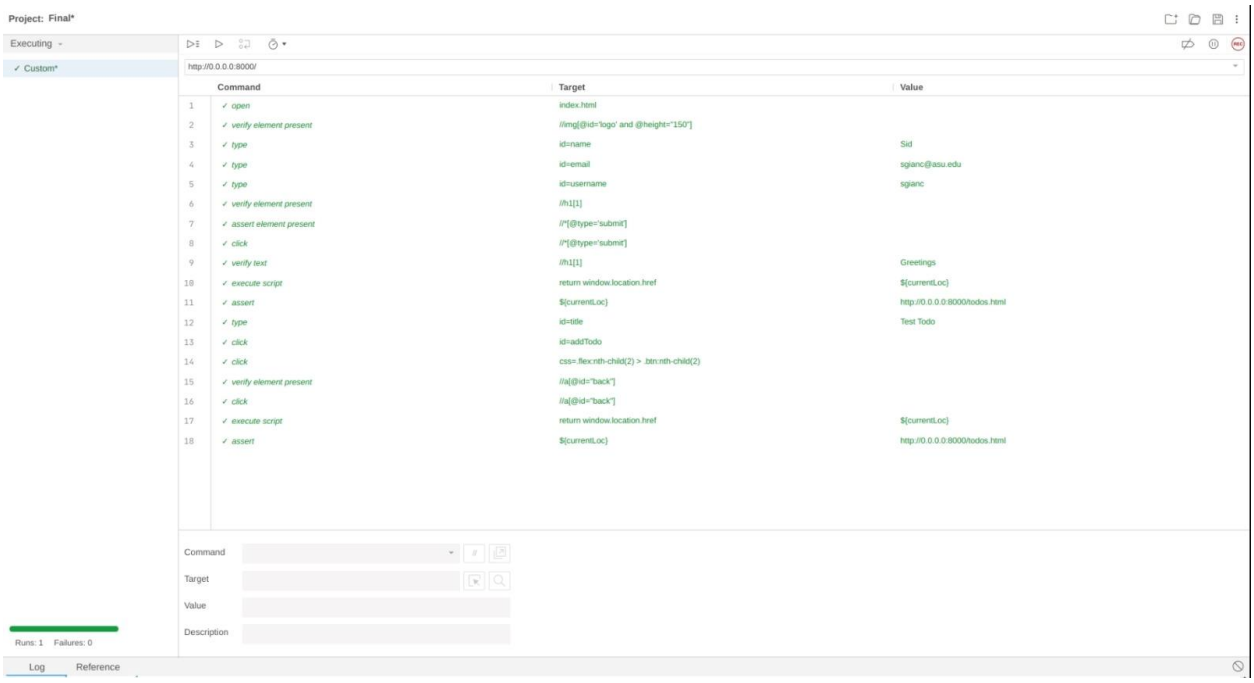
The button “Mark as Completed” in the second page has been changed to “Mark Completed” in version 2.

Change to test for correctness of link:

The link “Todo” for the third page is redirecting to the login page for version 2, instead of the second page as in version 1.

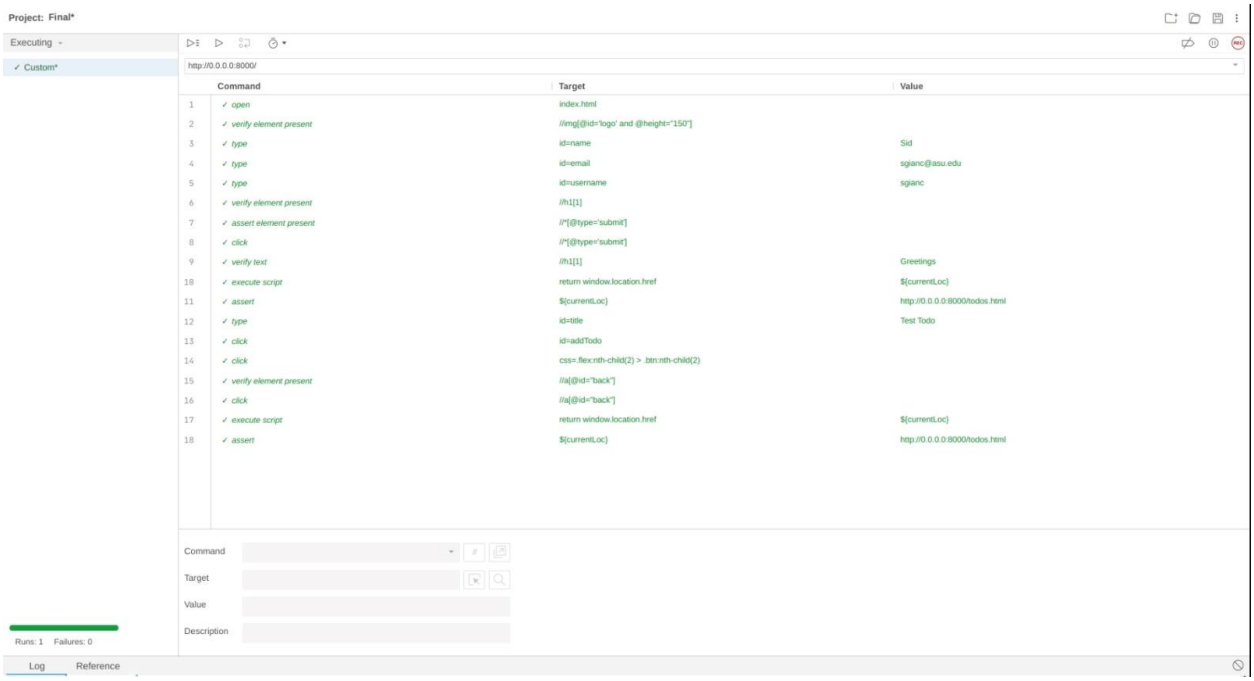
The test cases shown below were developed to test all these changes.

Test Cases for Version:

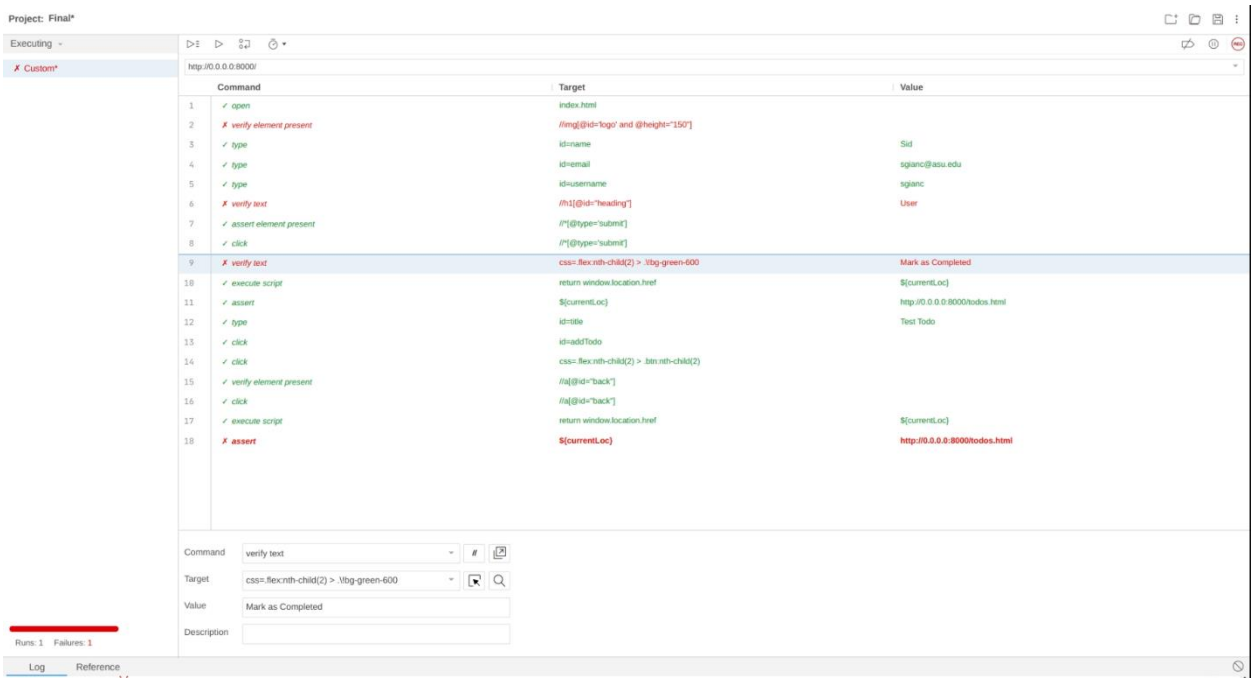


Explanation of test results with screenshots:

Test results for version 1:



Test results for version 2:



Explanation:

The test cases are failing for version two due to the changes mentioned in the previous section. For example, the test case for the location of “User” is failing for version 2. The test case of the

size of the image on the login page is also failing. The test case for the location of the label on the page for managing tasks is also failing. Additionally, the test case for the button “Mark Completed” will fail due to the name being different in version 1. Finally, the test case checking for the correctness of the link on the page for viewing each task in detail (that redirected to the page for managing tasks) will fail for version 2 where it is redirecting to the login page.

Assessment of tool [5]:

Features and Functionalities:

Selenium IDE has different features like a run button that runs a particular test, a run all button that executes all the test cases as well as a slider for changing the speed of test cases and test execution from slow to fast. The user can also pause the test case execution and then resume the testing from specific test cases. Additionally, Selenium IDE also aids in writing unique codes for scripts. There is also a feature called “Apply Rollup Rules” to condense all commands into one action to be executed.

Type of coverage:

Selenium IDE provides complete coverage of all the graphical elements that the tester interacts with when building a new test case. This includes testing for their existence, size, location, value and link correctness.

Reuse of test Cases:

Selenium IDE records all the interactions made between the tester and the graphical user interface. This means that it develops reusable automated test cases that can be used at any time. This helps reduce the time and effort required for testing new versions of software. It also makes test management easier and reduces repetition across test cases. The test cases developed by Selenium IDE can be reused regardless of any changes made for the new version and it will accurately show made errors are encountered in the new version, which helps enhance the quality of the product which will be released to the market.

Test Results Produced:

When running the test cases in Selenium IDE, we can view which cases are being processed, the warnings associated with them, which cases are passing and which ones are failing when we do not choose the logging option. Alternatively, after the final command is processed, we can view the summary of the test cases and their results in the logs by choosing the logging option. Selenium also shows us the number of successes and failures.

Ease of Usage:

Selenium IDE is very user-friendly. After giving it the url of the software and recording a testing scenario, it builds a test cases that will automatically launch the web browser and test the software. Since the url given can be changed at any time, this allows the test to be done for future versions without the need of recording a new video. Sine the test cases are easy to write and observe when being run, Selenium IDE becomes very easy and efficient to use. Additionally, Selenium IDE also provides concise and precise information about the tests which are failing, which help development efforts.

Type of GUI elements that can be Tested:

Various GUI elements that are in a webpage can be tested by Selenium IDE including check boxes, text boxes, labels, buttons, radio buttons, dropdown menus and images. Some of them have been included in my project and successfully tested for.

References:

[1]. BrowserStack, https://www.browserstack.com/guide/what-is-selenium-ide
[2]. Legacy Selenium IDE, https://www.selenium.dev/documentation/legacy/selenium_ide/