Choosing the best automation framework/tool depends on what are the requirements (e.g., tasks to be performed), what platforms to support (e.g., windows/mac/linux), and what kind of test is required. Based on the current requirements, I selected the automation tool **AutoIT** (https://www.autoitscript.com/site/autoit/) that was initially created to automate scripts for Microsoft Windows programs. While AutoIT is run, the result can be seen in real time, for example, opening excel and creating border in a cell can be seen by the user while the task is being performed. However, one major disadvantage of AutoIT is it only supports Microsoft Windows operating system. Therefore, it cannot be run on other platforms. One another cross platform framework is **Apache POI** (https://poi.apache.org/), which provides Java libraries for reading and writing files in Microsoft Office formats, such as Word, PowerPoint, and Excel. However, Apache POI is a backend tool, and while it is running the results are not visible.

I considered each scenario as separate, and excel file is opened and closed for each task.

My approach in listing the seven scenarios are presented below:

1. Launch the Excel and print the version number.

Launching excel is pretty straightforward. Initially, I thought to find the excel version by going through the menu. But this required sending a lot of commands; therefore, I used a query that was suggested in :

<http://support.moonpoint.com/os/windows/office/office_versions.php>

2. Switch between different tabs, and print Count of tabs.

This one was pretty straightforward. I got the worksheets in an array and iterated through it, and finally printed the number of tabs as well giving a message to the user.

3. User must be prompted with input box (accepts text of any tab names), and then focus must be automatically switched on to the specific tab.

This task was similar to task 2. The only difference was asking the user for a specific tab, and switching focused to it if it existed. Otherwise, an error message is printed and also a notification is provided to the user.

4. Print different control types (For Example Ribbon, list…) with in the Excel.

This task required to send a few keys. I did not want to use mouse coordinate since it might not be consistent across machines. From the options menu, ribbon list is accessed and the name of each item is stored in a local variable. Finally, at the end, it is presented to user.

5. Expand the Borders list, count the number of options in this list, and click on any of the border. Verify that selected border is applied.

Selecting border is not so difficult. It can be done through clicking in the border option in the list or using one line command to directly add the selected border with linestyle (I provided this option as comment).

6. Create two columns with desired number of records. Compare values in adjacent cell. Produce the report pass/Fail.

I compared each two column and the result is placed in the third column. I compared different scenarios, such as equal/unequal number, string, and an empty string.

7. Type any text with in the cell, read it and print the text, it could be either first few characters or middle characters.

This task was pretty straightforward. I wrote the text in a cell then read it and printed both a few characters and also the full text.