Lawzeem Latif

Documentation for Automation Test

1)

For part 1, to run the code locally, go to the Local directory and first run: “pip install -r requirements.txt”

If you are on windows, you can run the code directly from a terminal using python: “python Part1.py”

On any other platform you will first need to download chrome webdriver from https://sites.google.com/a/chromium.org/chromedriver/downloads.

To modify the search criteria, you can open the python script and change the Location, Adult, Children, Rooms, Start and End date as well as number of results displayed.

To run it in docker, go to the docker folder and run “sudo docker build -t part1” and after it is done, use “sudo docker run part1” to see the results.

2)

For part 2, install the requirements with “pip install -r requirements.txt” and then “python Part2.py”

The script will go to the website and to the files section and search the file type requested. It will then use python requests to download the file to the current directory.

In docker, follow the same build and run steps. However, the downloaded files will be in the docker container and you will need to copy them to the host machine.

3)

For part 3, the only requirement is selenium which can be installed using “pip install -r requirements.txt” and run using “python Part3.py”. The script will go to the given website and take screenshots of the requested links.

Inside the script file there is a python dictionary mapping link titles and their id’s. You can type in the titles you can screenshots of and the script will get them.

In docker, follow the same build and run steps. The screenshots will be in the docker container and you will need to copy them over to the host machine.

4)

Webdriver flags can be modified to prevent flagging by changing the driver options. User-agents can be rotated. IP proxies can be used as well.

Additionally, there are some experimental options that can be toggled like useAutomationExtension to prevent websites from identifying automated scripts.

Screen size can also be adjusted to make the request appear as if it is coming from a regular source instead of some non-standard resolution.

5)

Lawzeem Latif

It is unlikely that automated testing will replace manual testing because there are a lot of small changes that can be made which bots will not be able to handle. Additionally, a completely comprehensive automated test will be expensive to develop and maintain and will need regular adjustments.

Automated tests can complement manual testing to give developers with front-end and back-end knowledge more insight as they understand why the code does what it is doing and they can use that to make any changes needed.