Zense Recruitment Project Report

For Project Made by
Aravind Veluri
IMT2018010

Idea:

This project focuses on automating the tasks that a user performs on YouTube and on scraping the information of top video results for a given out by a YouTube search.

Technologies and Modules used and Implementation Details:

The program is written is Python 2.7 language. Selenium module in python, Mozilla Firefox browser and Firfox webdriver(geckodriver) are the technologies used in this project.

The python script has functions defined in order for specific purposes. All the functions take driver argument for reusability of code with another browser or new instance of same browser. The function "YouTubeHomepage" takes driver agrument and opens a YouTube page using "get" function in webdriver instance. Next function, "YouTubeSignIn" takes driver argument, finds the text "SIGN IN" in an <a> tag of HTML and clicks on it. Next, it finds the Gmail ID <input> tag by name "identifier" and sends the username inputted in terminal and clicks enter key to submit. Next, after loading the next page, it finds password <input> tag by name "password" and sends inputted password by the user and submits it. "time.sleep" has been implemented to let the browser load content, and let user comprehend what's happening in the browser.

Next function, "toggleDarkMode" takes driver agrument, clicks the avatar button after finding the tag by ID, prints the name of the user, clicks the dark mode button after finding the tag it by class name, and clicks the toggle button after finding it by ID, and comes back to the homepage by clicking back button and avatar button again. Next, Two lists "subbedChannels" and "unsubbedChannels" have been declared to store the name of the YouTube channels that have been subscribed to and unsubscribed from in the runtime of this script.

Two functions, "subscribeChannel" and "unsubscribeChannel" have been implemented, which take 2 arguments, driver and a string which has url of the channel, to subscribe and unsubscribe from given channel. Both functions find the subscribe button by xpath of the subscribe element, and check the value of the attribute "subscribed".This is where the both functions differ. This attribute returns None if the user is unsubscribed from that channel and return empty string if the user is subscribed to that channel. Based on this, if statement is implemented in both, and this attribute is checked, and then both functions click the button depending on this attribute. The unsubscribeChannel function additionally confirms the unsubscription which is not needed in subscibeChannel. Both the functions print if the channel is already (un)subscribed or not.

The function "YouTubeSearch" takes 2 arguments, a driver and a string to search in YouTube. This function finds the search <input> tag of YouTube and send the argument string to the input tag and submits it.

Next, the function "topResults" takes 2 arguments, a driver and a string to search in YouTube. First, it uses YouTubeSearch function to search the string on YouTube, then seperate lists for storing urls, channel name, video views, uploaded time for each video. Next, video elements are stored in a list by searching for the common tag which the result videos have, from parent element which stores all the video elements. A for loop is implemented on video list to get all the required data that we need(url, channel name, views, upload time) from a video by finding the the corrsponding elements and append it into approriate lists. WebDriverWait is used to let the page load all the required elements instead of directly searching for elements which haven't yet loaded resulting in a NoSuchElementException. This function returns a tuple which is a zip function call with 4 arguments as above given lists.

The program prints the zipped list, and asks the user to input a number between 1-10 which corresponds to the video in the zipped list(in order) and after taking the input, it plays the video.

Finally, two functions Like and Dislike which take driver argument, are implemented, which checks if the the video is playing or not, and then they find the like button by (absolute)xpath, and click them, if they are not already clicked. The program asks the user for input if they liked or disliked the video, for which the user should input 'yes' if the user liked it, 'no' if they disliked it and just press enter key, if the user wishes to remain neutral.

I also added the option of enabling and disabling extensions when opening a web browser using a script.

Video Links:

Youtube Link:

https://www.youtube.com/watch?v=r5rQkx4gb2Y&t=2s

Alternative Google Drive Link:

https://drive.google.com/file/d/1uUcMqGssaRkMEISR3csjKF9GMjMiTNn9/view?usp=sharing

Future Scope:

I would like to implement all these function using Object Oriented approach to make the code reusable. I would add the feature of adding comments to videos, and I would learn and implement event handling in Selenium to act accordingly to events happening in the browser. I would also like to add the feature of uploading videos to YouTube to make the job of the YouTube content creator easy, and recording live video on YouTube. I would also learn the required flash object web driver and JavaScript to add options for the user to control the video such as playing, pausing, muting, going full screen, etc. I would also like to learn machine learning and apply it here so as to filter clickbait

or false/useless videos by reading comments or examining the thumnail and removing them out of the search results.

My Experience:

As someone who spends quite a bit of time on YouTube, I really enjoyed doing this project and learning how YouTube's HTML code works. I learnt about HTML, CSS tags, attributes, values and how they dynamically change, their usage and how they work while doing this project. Finally, I learnt a new technology, Selenium, and how to automate stuff in web browsers using selenium.