UnSpam : Unsub the Spams

Introduction:

"Receiving a spam message—even if you do not open it—has an environmental impact of 0.3 gCO2e.The global carbon footprint from spam annually is equivalent to the greenhouse gases pumped out by 3.1 million passenger cars using 7.6 billion litres (two billion gallons) of gasoline in a year."

~Scientists at McAfee

"They are irritating"

~A layman

Spam mails are always on the top of the hate list of people who regularly use their email. And after some searching I found out that they're harmful to the environment in a very severe way. So one way to get rid of this was to manually delete all the spam mails, but it was not a complete solution. So I decided to build something which will unsubscribe the mails already identified as spam by the email service. So this is the aim of my Project.

Technical and implementation details:

I have used Selenium framework to automate the process of unsubscribing the spam emails. The whole code was written using Python 3. It involved many steps:

1. **Logging in**: I have used Gmail to do my work for now. I have to enter the email id and password in the terminal and the script logs in automatically. It stops to ask for One Time Password(OTP) if 2-step verification has been activated. I have used the getpass module to take the password input, this doesn’t display the password when it is being typed into the terminal.
2. **Finding the Spam Folder**: Here I basically just find the Spam Folder button using the XPath in the source code. This helped me learn the basics of XPath and how to use it with Selenium.
3. **Accessing the Emails**: This was probably the most interesting part for me as I learned here that all the emails are individual rows of a HTML Table Element and can be easily accessed as they’re indexed by default
4. **Finding the Unsubscribe Link**: With people finding ways to avoid spams, spammers have found out ways to get around these measures. I have only tried to find links using a few key words and this works for majority of cases but not all of them. In selenium we can automate stuff by using “drivers” which are basically different tabs/windows. Once we click a link we lose control over it, as it opens in a new tab. So to avoid this I used 2 drivers, one for the mail and another for the Unsubscribe Link.
5. **Completing the Unsub**: After using the second driver, there are few more problems. Not all pages behave in the same way, few require us to select reasons from a

drop down, few require us to check a few boxes and some just provide a simple button. So this is something I want to improve on as I haven’t done it for a more real life situation. This part is not completely done.

Future scope:

My main aim would be to make the script find the required stuff in a more efficient and intelligent manner. I plan on using Machine Language in the future to help with this process. The last part of the script is something I want to improve a lot. Also making this available to many more email services is another aim. Using this script might help reduce the number of unread emails and in one way reduce memory usage.

Overall Experience:

After this project I learned how the ‘inspect source’ process works. Using the web to solve my errors is one good thing I could practise. I faced a situation with a deadline for the first time where no-one is available to help as everyone is doing there own project. This was one thing I need more of. Thinking of an idea was a very difficult task for me as I thought of everything to be very easy even before doing it. But once I started , the same easy thing started causing problems. Learning facts like how emails are stored in HTML and how to access everything on a webpage kept me involved but the fact that this project couldn’t be as good as what I wanted it to be was a lesson regarding proper planning and knowing what you can do.

Video Link: https://youtu.be/ZuPL1rNOgPI

GitHub Link: https://github.com/sal2701/UnSpam