Regular Expressions Assignment

Use Python and the Regular Expressions package to find as many instances of the data referenced in each row of the table below from the Joint Hearing transcript on "FACEBOOK, SOCIAL MEDIA PRIVACY, AND THE USE AND ABUSE OF DATA." That document, which can be found at

https://www.congress.gov/event/115th-congress/senate-event/LC64510/text?s=1&r=59

or more conveniently,

https://www.congress.gov/115/chrg/CHRG-115shrg37801/CHRG-115shrg37801.pdf,

was converted to a text document with the filename `CHRG-115shrg37801.txt` for more convenient use with Python.

Write a Python code file in Spyder entitled `regex_hw.py` that creates a list of the results found for each data target listed in of the rows in the table on the next page. Use the variable name mentioned in each row for the respective list. Please retain duplicate names within each list. Furthermore, follow these specifications in writing and submitting your code:

- Write the program so that it looks for the data file (above) in the same folder where your `regex_hw.py` code resides. That is, do not include any path specification beyond the name of the input file name.
- Submit your code file to your network drive as you did for the webscraping assignments.

You may use online generative AI engines in this assignment to do the following ***after*** you have written a first version of your program and Regex statements:

- To generate alternatives to help you develop better Regex commands, which you may copy and include in your code if they are superior to your initial draft code.

Include these comment statements in your code to, first, indicate, whether you used Gen-AI and, second, how you used it and whether it was useful.

Did you use GenAI: Yes or No

‘’’ How did you try to use GenAI and how useful was it? Were chatGPT's suggestions better than you initial draft?

‘’’

Note that you will need to type the triple quotes for the multi-line comment into Spyder because copying and pasting is not likely to copy the single quote character correctly.

Hints:

- Don’t worry about your Regex patterns being perfectly specified. Prioritize simpler statements rather than perfect ones, even if they sacrifice a little precision. Writing perfect Regex patterns is difficult and this is, for many of you, your first experience with Regex.
- If you Regex code does not return perfect results, remember that you can edit your results using basic Python string methods such as stripping unwanted text from the left or right sides of the results, substituting text, and so forth.

Regex Searches to be Conducted

Item	Example(s)	Python Variable	Note
Internet URLs	`https://www.nytimes.com/2018/03/27/`	`url`	• URLs may start with `http://` or `https://`
Email addresses	`info@globalscienceresearch.com`	`email`	Use these characteristics of email addresses, although these are not exhaustive specifications: • These special characters are allowed in email addresses in addition to letters and numbers: !#$%&'*+-/=?^_`{|}~ • The components of an email address before and after the @ may have multiple groups of letters separated by periods. • An email address must have one and only one @ character.
Speaker Names	`Ms. MARTORANA., Mr. Zvenyach., Mr. ZUCKERBERG.`	`name_spkr`	These names appear at the beginning of statements and at the beginning of each line where a speaker is quoted. The names are terminated with a period. It is okay to include the period in the captured text. That said, it is easy to strip the periods from a list generated by Regex if necessary.
Names of Representatives and Senators who presented statements	`HON. TED CRUZ, HON. CATHERINE CORTEZ MASTO`	`name_rep`	Some titles include 2 names whereas some have 3 names.